INFORMATION HANDLING PROPERTIES OF NEUROMIME NETS

ROBERT M. COLOMB

SYSTEMS RESEARCH LABORATORIES, INC.

SEPTEMBER 1966

AEROSPACE MEDICAL RESEARCH LABORATORIES
AEROSPACE MEDICAL DIVISION
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO

NOTICES

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Requests for copies of this report should be directed to either of the addressees listed below, as applicable:

Federal Government agencies and their contractors registered with Defense Documentation Center (DDC):

DDC
Cameron Station
Alexandria, Virginia 22314

Non-DDC users (stock quantities are available for sale from):

Chief, Storage & Dissemination Section
Clearinghouse for Federal Scientific & Technical Information (CFSTI)
Sills Building
5285 Port Royal Road
Springfield, Virginia 22151

Organizations and individuals receiving reports via the Aerospace Medical Research Laboratories' automatic mailing lists should submit the addressograph plate stamp on the report envelope or refer to the code number when corresponding about change of address or cancellation.

Do not return this copy. Retain or destroy.

800 - January 1967 - C0192-17-332

INFORMATION HANDLING PROPERTIES OF NEUROMIME NETS

ROBERT M. COLOMB

FOREWORD

This report covers the work done by Systems Research Laboratories, Inc., Dayton, Ohio, in development of theoretical models of biological information handling systems. This effort was performed under Contract AF33(615)-1825, beginning 15 May 1964 and ending 31 May 1965. This work was performed for the Aerospace Medical Research Laboratories and administered by the Biomedical Laboratory in support of project 7233, "Biological Information Handling Systems and their Functional Analogs," task 723302, "Biological Servomechanisms." The technical monitor was Lt. Colonel Jack E. Steele, MC, USAF, Mathematics and Analysis Branch, Biodynamics and Bionics Division, Biomedical Laboratory.

The author wishes to express his appreciation for the assistance of the following members of the SRL Staff: Mr. Larry Irwin for contributing to the inception of and reviewing the manuscript of Appendix III, Mr. William Markel for assistance in preparing Appendix IV, and Mr. Thomas Rauterkus for contributing the material in Appendix V.

This report has been reviewed and is approved.

J. W. HEIM, PhD
Technical Director
Biomedical Laboratory
Aerospace Medical Research Laboratories

ABSTRACT

This report is a study on some elementary information handling properties of neuromime nets, giving most emphasis to the functioning of a single neuromime component, and containing some discussion of the operation of simple nets. Single component computation is treated from the point of view of changes brought about in the internal structure by operations performed during data flow. A geometrical model is presented which illustrates the pattern measurement behavior of the component, and some of the simpler differential equations of adaptation are solved to provide some insight into the effect and interaction of the component control parameters. Simple net behavior is concerned mainly with feedback interaction among components, and gives some useful notation for describing net operation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES (Cont)

LIST OF TABLES

Section I

INTRODUCTION

This report is a description of the computations performed by a type of neural net, and can function as a programming manual for this net.

The contract under which this report was prepared is a part of an effort to develop more effective information handling systems, in the sense of extracting usable knowledge from received signals.

More generally, the overall effort is concerned with the development of information processing systems with capabilities approaching those of the human brain. The approach being taken to solve this problem is to analyze the behavior of the brain, breaking it down into functional components, then analyzing each component until a basic component is reached which can be constructed with present technology. These components are then constructed (or simulated) and put together in various systems, and the behavior of these systems compared with actual cognitive operations.

Emphasis has been placed on the functioning of individual components and simple nets. Section II gives a basic description of the operations performed by components, the kinds of net structures the component is designed to form, and a general discussion of the computational parameters. Section III describes informally the computations performed by a single neuromime, with particular attention to adaptation. Section IV is a more rigorous discussion of the computational processes of a component, section V of the adaptation process, and section VI discusses simple nets. Section VII considers a number of extensions, conjectures and unsolved problems that have arisen in connection with this research.

Section II

GENERAL PROPERTIES OF NEUROMIME AND NETS

The problem under consideration is that of building a flexible pattern-recognizing device to be used in decoding signals produced by some sort of sensory encoders. Instead of producing a device directly, it is desired to investigate the design of a more general computation system that would be self-organizing.

This report is an analysis of some of the mathematical properties of a computation system developed by the contract monitor.* The system is modeled after an organizational mechanism shown in Figure 2.1. There are a series of computation areas with a number of transmission lines connecting them with each other, and with the outside, both input and output. The computer would have the ability to alter the computation parameters of an area in such a way as to alter its responses with time and it would also have the property that some areas could be activated and others de-activated for periods of time.

A computational element, called a neuromime, has been designed for the system and is illustrated in Figure 2.2. There are four sets of 10 input lines each, called $\vec{I}^+$, $\vec{I}^-$, $\vec{S}^+$, and $\vec{S}^-$. Each set can be regarded as a 10 dimensional vector. Since all transmission to and from the element is in the form of positive numbers, each element in each of the vectors is positive.

Associated with each input vector is a weighting vector, respectively $\vec{G}_{I+}$, $\vec{G}_{I-}$, $\vec{G}_{S+}$, and $\vec{G}_{S-}$. The output R is computed as follows: [B is a bias--used as the exciting or inhibiting mechanism, M_I and M_S are weighting factors]

$$
\begin{aligned}
I &= B + M_I(\vec{I}^+ \cdot \vec{G}_{I+} - \vec{I}^- \cdot \vec{G}_{I-}) \\
S &= M_S(\vec{S}^+ \cdot \vec{G}_{S+} - \vec{S}^- \cdot \vec{G}_{S-}) \qquad (2.1) \\
R &= \max(I + S, 0)
\end{aligned}
$$

*See Neuron Component Development, Semi-Annual Report, Contract AF33(616)-6805, General Electric Co., 15 June 1960; Gracer and Orr, Neuromime Network Simulation, Final Report, Contract AF33(657)-11194, Service Bureau Corp., 14 August 1964; Neural Network Simulator, Final Engineering Report, Contract AF33(657)-8489, Teledyne Systems Co., January 1965.

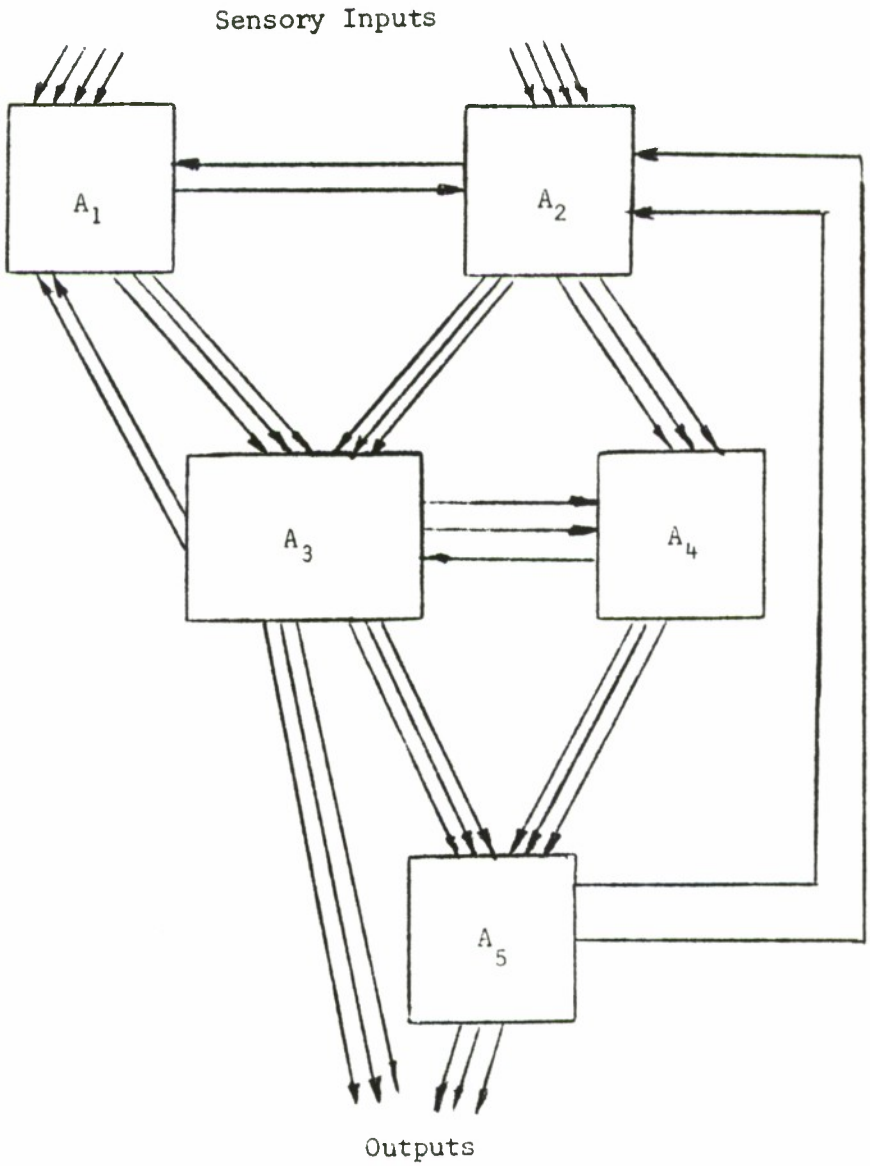
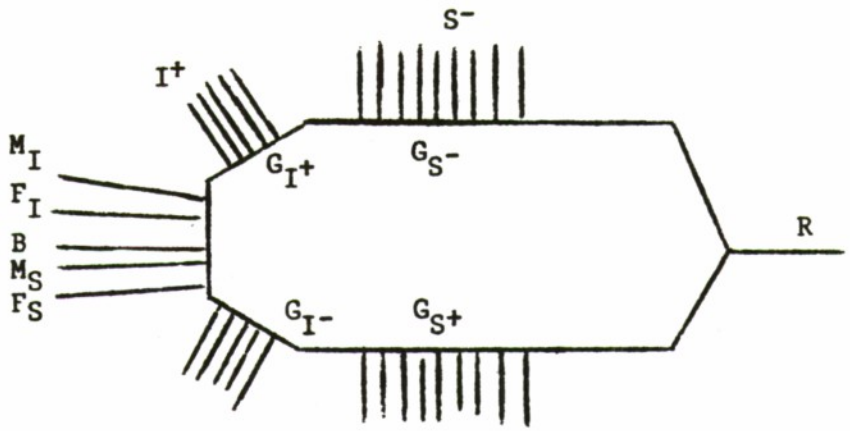


Figure 2.1. Computational Structure for Self-organizing Computer Showing Transmission Lines and Computational Areas



Logical Diagram of the Steele Neuromime

Figure 2.2

After the computation of R, the G's are altered to make them respond better to the input signals. [F_I, F_S are weighting factors, $\vec{\bar{A}}$ is a vector with the same number of elements as $\vec{A}$, each equal to the average of the elements of $\vec{A}$.]

$$\begin{aligned}
\vec{G}_{I^+} &= F_I \cdot R \cdot (\vec{I}^+ - \vec{\bar{I}}^+) + \vec{G}_{I^+} \\
\vec{G}_{I^-} &= -F_I \cdot R \cdot (\vec{I}^- - \vec{\bar{I}}^-) + \vec{G}_{I^-} \\
\vec{G}_{S^+} &= F_S \cdot R \cdot (\vec{S}^+ - \vec{\bar{S}}^+) + \vec{G}_{S^+} \\
\vec{G}_{S^-} &= -F_S \cdot R \cdot (\vec{S}^- - \vec{\bar{S}}^-) + \vec{G}_{S^-}
\end{aligned} \tag{2.2}$$

subject to some restraint on the final value of each G vector. One restraint considered is that the sum of the elements in each G vector

should remain the same, another is that the sum of squares of the elements in each G vector should be constant.

A computational area in the computer is composed of a number of these elements. Inputs to the area from other areas or from the outside are connected to I inputs of the neuromimes. The S inputs are connected to the outputs of neuromimes in the same area to give feedback to the system. The inputs B, F and M can be from anywhere in the system.

The overall combination of computational areas is called a net, and a single area is called a level. The organization of a level is illustrated in Figure 2.3. The I inputs to each component are taken from external sources, while the outputs of some of the neuromimes are connected to the S inputs of others. All S inputs in the level are taken from outputs of other components in the same level. The external sources will be termed the Input Space.

Parameters affecting the computations of the level are M_I, M_S, F_I, F_S, and B. F_I and F_S are adaptation weights which have no immediate effect on a particular computation, while B is a bias acting on each component of the level. If $M_S = 0$, then each of the components performs a measurement on the Input Space independent of all other measurements in the level, except, of course, of possible overlap and statistical correlation between the inputs to the different components. When M_S is nonzero a degree of interaction is introduced.

These parameters are designed to yield the following behavior from a level:

1. The output of a particular component can be either increased or decreased by the output of a neighboring component.

2. The degree of interaction can be controlled on a global basis; that is, the general degree of dependence of the outputs of components on the outputs of other components can be controlled.

3. The degree of dependence of the output of a particular component on the outputs of its neighbors can be controlled on a local basis.

4. The overall activity of the net can be controlled.

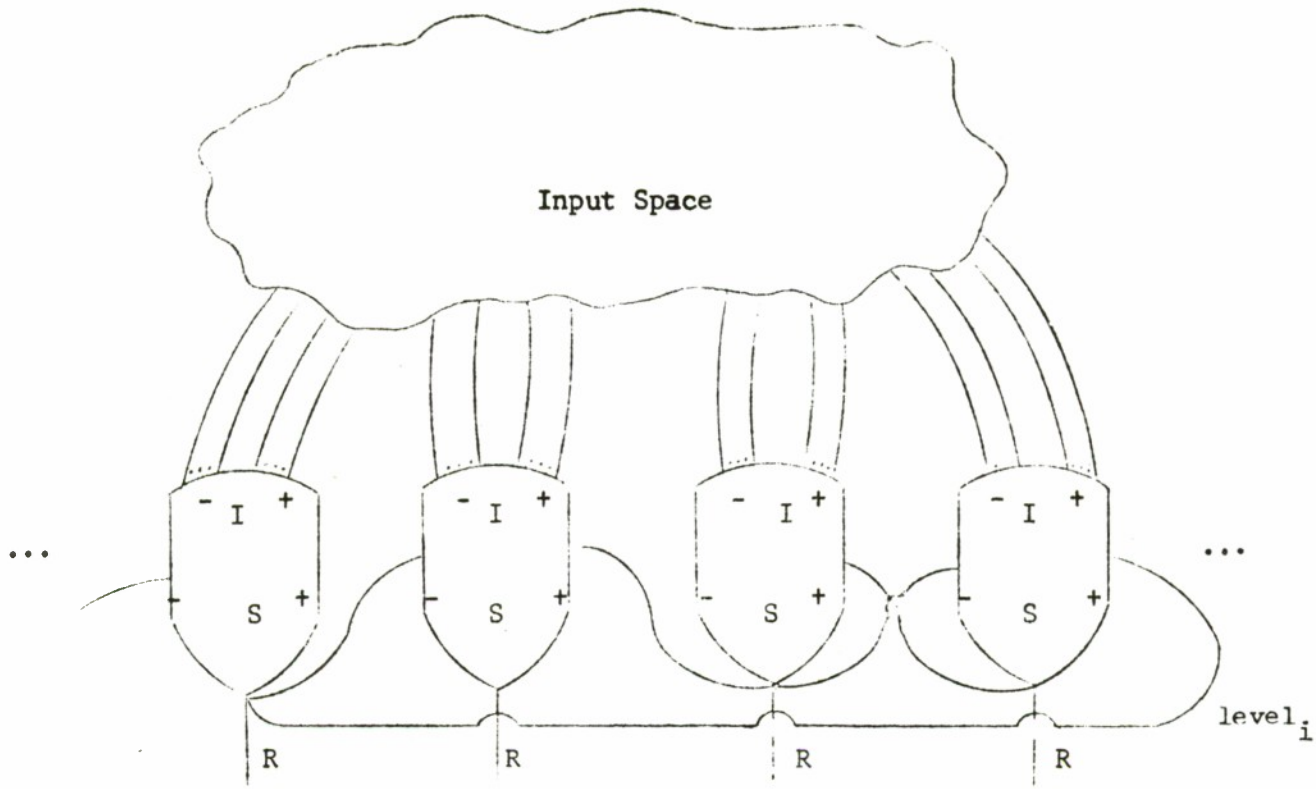


Figure 2.3. Example of Neuromime Net

Figure 2.4 shows how the first consideration is effected. The component C has as an S^+ input the output from component N_1, and as an S^- input the output from component N_2. A large output from N_1 will raise the output of C, and a large output from N_2 will lower the output of C. The degree of interaction is given by the ratio of M_I to M_S. If $M_I/M_S >> 1$, there will be little interaction, if $M_I/M_S << 1$, there will be a great deal of interaction.

M_I and M_S are parameters common to each component on a level. An adjustment in their ratio therefore increases or decreases the interaction generally.

It is necessary to make use of a special kind of interconnection to obtain the third consideration. In Figure 2.5, we see that the output of component N is connected to both the S^+ and S^- inputs of component C. The degree of interaction is thus dependent on the G_S set of C. The effect of N on C is

$$R_N g_i^+ - R_N g_j^- = R_N(g_i^+ - g_j^-) \tag{2.3}$$

where g_i^+ is an element of $\vec{G}_S^+$, g_j^- is an element of $\vec{G}_S^-$, and R_N is connected to the ith S^+ input and to the jth S^- input.

Initially, $g_i^+ = g_j^-$, so that there is no interaction. If F_S is nonzero, however, the G_S sets will be altered in the same manner as the G_I sets are altered with F_I. A positive F_S will increase g_i^+ and decrease g_j^-, making the interaction of N and C positive, and a negative F_S will have the reverse effect, thus altering the local degree of interaction.

The bias, B, which can be either positive or negative, is a device to adjust the activity threshold of each component in the level. If this bias is low, the level will tend to be "aware" of only those inputs for which it is set up to give a maximal response. Conversely, if the bias is high, the activity will be increased, and the level will tend to be "aware" of inputs which it formerly ignored (because $R \leq 0$)

For further discussion of the action of a level, it is useful to have in mind the mathematical formulations and geometric models developed

in the following chapters. Most of the work has been for the special case of constant inputs to a level, although in the last chapter some preliminary formulations are given for time varying inputs.

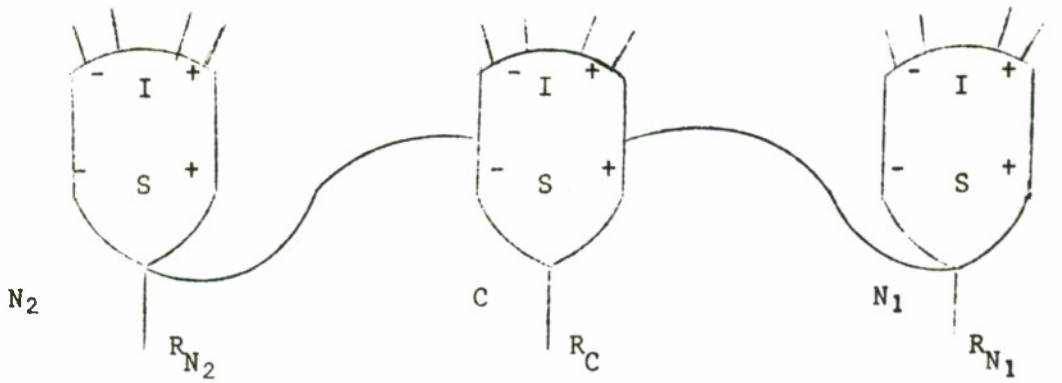


Figure 2.4. Sample Net showing Positive and Negative Interaction

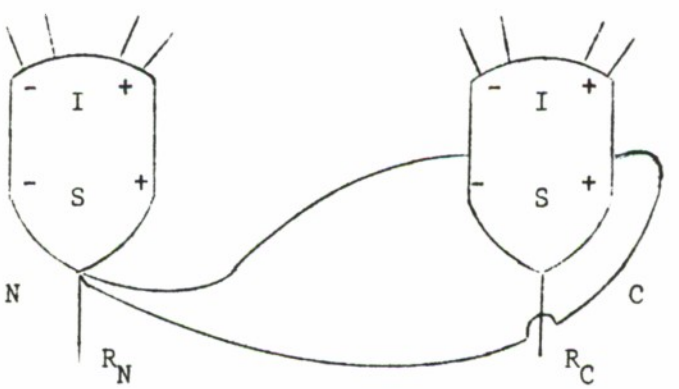
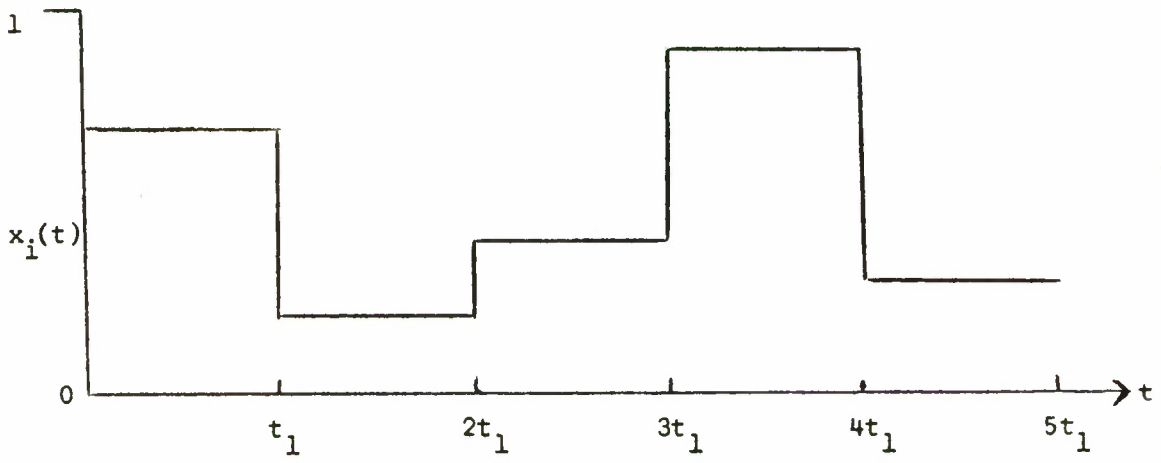


Figure 2.5 Example of Interaction Alterable to either Positive or Negative by G_S-set Adaptation

Section III

COMPUTATIONS PERFORMED BY A SINGLE NEUROMIME

A neuromime can be considered as an adaptive measuring device on an input space. Its input consists of $2n$ lines, n of which are attached to the I^+ inputs, and n to the I^- inputs. These lines can be considered as a $2n$-dimensional time varying vector function $\vec{x}(t)$, whose first n components are inputs to I^+ and the last n to I^-. The components of $\vec{x}$ will be denoted as $x_1,\ldots, x_n, x_{n+1},\ldots, x_{2n}$. For the present, we will consider $\vec{x}(t)$ to be constant over a period t_1, changing as step functions. A typical component might have values as in Figure 3.1. The values of the components are restricted to be between 0 and 1.



Typical Sequence of Values for a Single Input to a Neuromime

Figure 3.1

Within the neuromime are stored a set of weights corresponding to the inputs. These weights may be denoted by the vector elements $g_1,\ldots,g_n, g_{n+1},\ldots,g_{2n}$. Since in Equation (2.1) the last n g-elements are subtracted, it makes sense to consider in the following the vector g

$$\vec{g}(t) = \begin{pmatrix} g_1 \\ \vdots \\ g_n \\ -g_{n+1} \\ \vdots \\ -g_{2n} \end{pmatrix} \qquad (3.1)$$

For a beginning in the discussion we will consider the simplest case of the neuromime, with $M_S = 0$, $B = 0$ and $M_I = 1$. The computations given by (2.1) reduce to

$$R = \vec{x} \circ \vec{g}$$
$$R' = \max(R,0) \qquad (3.2)$$

The restriction on the g elements considered in this report is that

$$\sum_{i=1}^{n} g_i = \sum_{i=n+1}^{2n} g_i = 1 \qquad (3.3)$$

There is a further restriction on each g-element

$$0 \leq g_i \leq \frac{1}{p} \quad i = 1,\ldots,2n \qquad (3.4)$$

for a given integer p. This latter restriction means that a g-element can saturate so that it cannot be increased. Its effect may be seen by considering the adaptation process in Figure 3.2

Figure 3.2 gives an example of the adaptation for a particular input on the excitatory inputs only. Part a) shows the input elements. Part b) shows the deviation vector of Equation (2.2)

$$\vec{D}^+ = \vec{I}^+ - \vec{\bar{I}}^+ \qquad (3.5)$$

which is used to guide the adaptation. Part c) shows the g-set before adaptation. Note that g_6 and g_2 are saturated. Part d) shows the

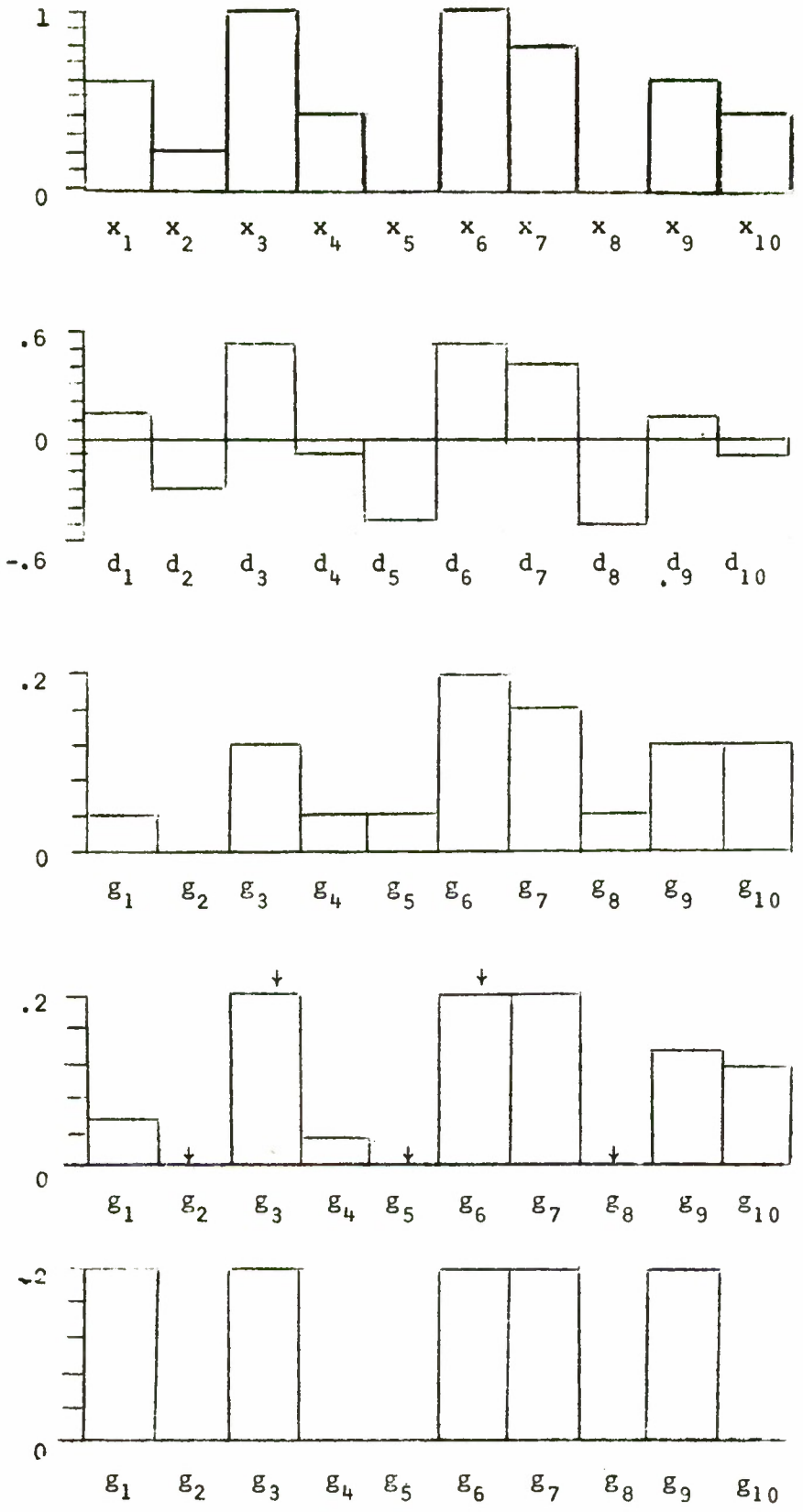


Figure 3.2. Example of Excitatory Adaptation with $F_I = .1$ and Showing Saturation

g-set after adaptation, with an indication of the g-elements which were unable to adapt by the full amount because of saturation. There is a mechanism in the simulator which adjusts for this occurence. Part e) shows the g-set as it would be if this input were repeated sufficiently.

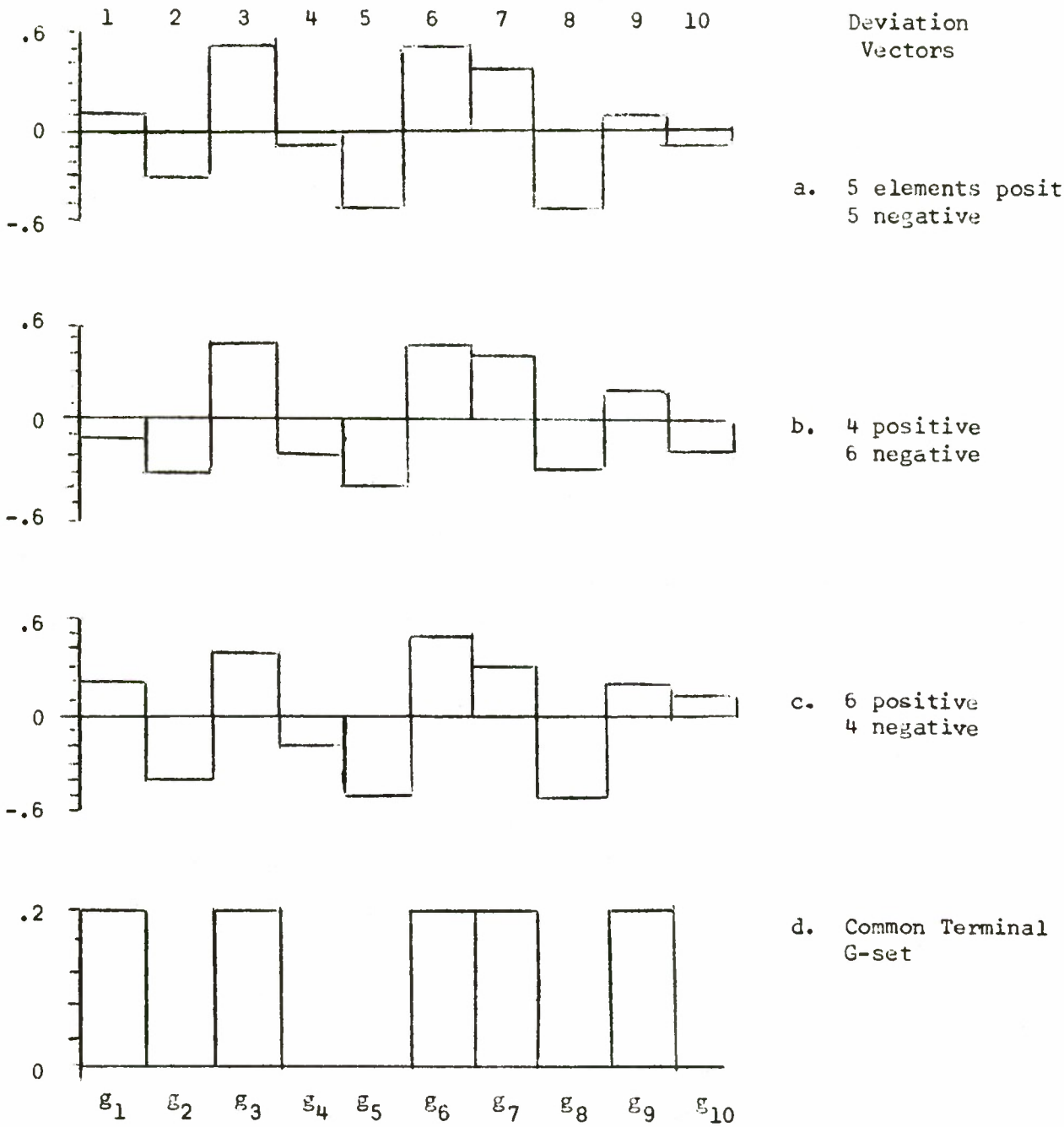
Using the deviation vector rather than the input vector in this adaptation is justified by the normalization criterion that the sum of the g-elements remain constant. The sum of the d-elements is zero, so that

$$\sum_{i=1}^{n} (g_i + F_I R d_i) = \sum_{i=1}^{n} g_i + F_I R \sum_{i=1}^{n} d_i = \sum_{i=1}^{n} g_i \qquad (3.6)$$

Note that in Equation (2.2), the inhibitory deviation vector is subtracted from the inhibitory g vector.

The terminal g-set is of interest because it in a certain sense picks out those inputs of most significance and ignores the rest. On the excitatory side it picks out the p largest inputs (those p which have the least negative deviation), and on the inhibitory side it picks out the p smallest inputs (those which have the least positive deviation). The component will give an output only if most of the significant excitatory inputs are present and most of the inhibitory inputs are absent. Figure 3.3 shows three deviation vectors which result in the same terminal g-set.

In practice, the g-sets will rarely achieve their terminal state, but will fluctuate about some intermediate configuration. In the operation of the net, the first input will generally find a neuromime for which it gives a high response. This neuromime (neuromime A) will adapt its g-set more toward this input than will any other, even though many neuromimes will produce an output and will adapt to a degree. For subsequent inputs which differ from the first, neuromime A will generally give a small output and will adapt toward this new input relatively little, while others will adapt more. If the first input recurs reasonably soon, neuromime A will have an even larger output relative to the others than before, and will continue this differential adaptation. The first input, however, will probably not recur often enough to drive neuromime A to a terminal state.



Three Deviation Vectors Resulting in the Same Terminal G-set

Figure 3.3

Section IV

GEOMETRICAL MODEL OF COMPUTATIONS PERFORMED BY A SINGLE NEUROMIME

Because of the vector nature of the operations in the neuromime, it makes sense to look upon them in a geometrical model. The adaptation restrictions given by equations (3.3) and (3.4) indicate that this model is related to a higher dimensional octahedron. A discussion of the formulation of this concept and an introduction to the notation and language of this model is given in Appendix I. The following discussion assumes that the reader is familiar with this material.

Since the terminal state of a neuromime is relatively simple computationally, we will develop the model from it. We will assume that the normalization is octahedral, as in equations (3.3), (3.4) and will consider only the I^+, I^- inputs by setting $M_s = B = 0$. We will speak of the G-set as divided into two vectors G^+ and G^- with elements $\{g_i^+\}$ and $\{g_i^-\}$, respectively. Each of the vectors I^+, I^-, G^+, G^- will have n elements.

The condition that the G-set is terminal means that each g_i^+, g_i^- equals either 0 or $\frac{1}{p}$, meaning that there are only p nonzero elements in each of G^+, G^-, denoted by $\{g_{i_1}^+,\ldots,g_{i_p}^+\}$ and $\{g_{j_1}^-,\ldots,g_{j_p}^-\}$ respectively. The computation of the neuromime from (2.1) reduces to

$$
\begin{aligned}
I &= M_I\left[\sum_{\ell=1}^{p} g_{i_\ell}^+ x_{i_\ell} - \sum_{\ell=1}^{p} g_{j_\ell}^+ x_{j_\ell}\right] \\
&= M_I\left[\sum_{\ell=1}^{p} \frac{x_{i_\ell}}{p} - \sum_{\ell=1}^{p} \frac{x_{j_\ell}}{p}\right] \qquad (4.1) \\
&= \frac{M_I}{p}\left[\sum_{\ell=1}^{p} x_{i_\ell} - \sum_{\ell=1}^{p} x_{j_\ell}\right]
\end{aligned}
$$

We can further assume that the sets $\{i_\ell\}$ and $\{j_\ell\}$ have no numbers in common, so that we can consider the sum above to be

$$I = \frac{M_1}{p}\left[\sum_{\ell=1}^{p} x_{i_\ell} - \sum_{\ell=p+1}^{2p} x_{i_\ell}\right] . \qquad (4.2)$$

Since the output R is computed by

$$R = \max(I, 0),$$

we can say that

$$R > 0 \quad \text{if and only if} \quad \sum_{\ell=1}^{p} x_{i_\ell} > \sum_{\ell=p+1}^{2p} x_{i_\ell} . \qquad (4.3)$$

The Input Space is a 2n-dimensional Euclidean space, restricted to the principal section, and the points for which $R > 0$ are divided from those where $R = 0$ by the hyperplane

$$x_{i_1} + \cdots + x_{i_p} = x_{i_{p+1}} + \cdots + x_{i_{2p}} \qquad (4.4)$$

which is a division of the principal section into two regions. If we restrict the inputs so that

$$\sum_{\ell=1}^{2p} x_{i_\ell} \leq 1 \qquad (4.5)$$

we find that we have restricted the inputs to lie in the interior and surface of 2n-dimensional octahedron. Furthermore, we can restrict our attention to that 2p-dimensional octahedron, Q, determined by the vertices $v_{i_1},\ldots,v_{i_{2p}}$, which is also divided by the (2p-1)-dimensional hyperplane, $H_0 = df$

$$x_{i_1} + \cdots + x_{i_p} = x_{j_{p+1}} + \cdots + x_{i_{2p}} \qquad (4.6)$$

or

$$x_{i_1} + \cdots + x_{i_p} - x_{i_{p+1}} - \cdots - x_{i_{2p}} = 0.$$

We will denote by A the region where $R > 0$, and by $\bar{A}$ the region where $R = 0$.

In the region A, R considered as a function

$$R: \; I^+ \times I^- \times G^+ \times G^- \rightarrow (0, 1]$$

defines a set of equivalence classes, one for each possible value of R. Geometrically, this equivalence class is a (2p-1)-dimensional hyperplane H_y:

$$x_{i_1} + \cdots + x_{i_p} - x_{i_{p+1}} - \cdots - x_{i_{2p}} = y \quad \text{for} \quad 0 < y \leq 1. \tag{4.7}$$

H_y is parallel to H_0.

An example of this for the 4-dimensional case is given in Figure 4.1. The figure shows the intersection of the 3-dimensional hyperplanes H_0 and H_1 with the face of the octahedron. The limiting case is the 2-dimensional hyperplane $x_1 + x_2 = 1$, $x_3 = x_4 = 0$, which is a line in the x_1x_2 coordinate plane. This is shown as H_1 in the figure.

For the 2p-dimensional case, H_1 is given by the equations

$$x_{i_1} + \cdots + x_{i_p} = 1, \quad x_{i_{p+1}} = \cdots = x_{i_{2p}} = 0 \tag{4.8}$$

and is thus a (p-1)-dimensional hyperplane, a degenerate member of the (2p-1)-dimensional family H_y. As such, it is also parallel to H_0.

We will denote the vector $(x_{i_1},\ldots,x_{i_{2p}})$ as $\vec{x}^*$. We can thus interpret R as a "distance" measure of the hyperplane containing the input point $\vec{x}^*$ from a reference hyperplane H_1. R will be 1 when $\vec{x}^* \in H_1$ and 0 when $\vec{x}^* \in \bar{A}$. A response R = 1 does not identify the input as a particular point, but restricts it to a (p-1)-dimensional hyperplane. In the example given in Figure 4.1, points (1, 0, 0, 0) and (0, 1, 0, 0) are both contained in H_1. An interval $0 < a \leq R \leq b \leq 1$ (interval (a, b)) defines a neighborhood in the space Q as the 2p-dimensional volume bounded by the (2p-1)-dimensional hyperplanes H_a and H_b. This is a kind of "slice" of the octahedron Q, so that two points can be in the same neighborhood even though their Euclidean distance from each other might be very large, and can be in different intervals even though it might be small.

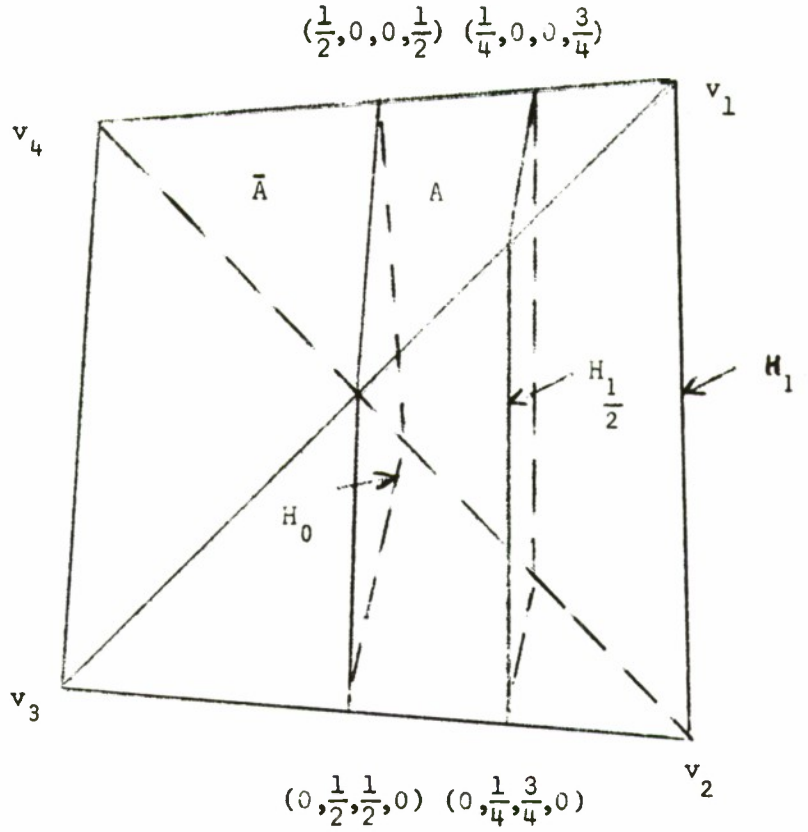


Figure 4.1 Four-dimensional Case--Intersection with Face of $x_1 + x_2 - x_3 - x_4 = y$ showing H_y for $y = 0$, $y = \frac{1}{2}$, $y = 1$

The more complicated non-terminal case can now be treated. Equation (2.1) now reduces only to:

$$R' = M_I\left[\sum_{i=1}^{n} g_i x_i - \sum_{i=n+1}^{2n} g_i x_i\right] .$$

Setting $R = R'/M_I$, we have

$$R = g_1 x_1 + \cdots + g_n x_n - g_{n+1} x_{n+1} - \cdots - g_{2n} x_{2n} = \vec{g} \cdot \vec{x} \qquad (4.9)$$

In addition, $\vec{g}$ is altered after the computation as follows:

$$\vec{g} = \Gamma \cdot R \cdot \vec{d} + \vec{g} \qquad (4.10)$$

where

$$\vec{g} = \begin{pmatrix} g_1 \\ \vdots \\ g_n \\ -g_{n+1} \\ \vdots \\ -g_{2n} \end{pmatrix} \qquad \vec{d} = \begin{pmatrix} d_1 \\ \vdots \\ d_n \\ d_{n+1} \\ \vdots \\ d_{2n} \end{pmatrix}$$

$$d_i = x_i^+ - \bar{x}^+ \quad , \qquad i = 1,\ldots,n$$

$$d_i = x_i^- - \bar{x}^- \quad , \qquad i = 1,\ldots,n$$

and Γ is a constant parameter. For the present, we can take $\Gamma = 1$. An example of this computation iterated until the G-sets become terminal is given in Table 4.1 for $n = 2$ and $p = 1$.

If both the g_i and x_i are restricted as before to the interval [0, 1] equation (4.9) for $R = 0$ determines a (2n-1)-dimensional hyperplane, H , dividing the principal section of a 2n-dimensional octahedron. This hyperplane always contains the line $x_1 = x_2 = \cdots = x_{2n}$ since

$$g_1 + \cdots + g_n = g_{n+1} + \cdots + g_{2n} = 1$$

by the octahedral normalization; and for

$$x_1 = \cdots = x_{2n} = a,$$

$$g_1 a + \cdots + g_n a - g_{n+1} a - \cdots - g_{2n} a = a(g_1 + \cdots + g_n - g_{n+1} - \cdots - g_{2n})$$

$$= a(1 - 1) \equiv 0, \qquad (4.11)$$

for any a. The intersection of H with the $2n^{th}$ order face therefore contains the point $x_1 = \cdots = x_{2n} = \frac{1}{2n}$. The line $x_1 = x_2 = \cdots = x_{2n}$ will be denoted by L.

The orientation of H can be determined by considering the intersections of H with the 2^{nd} order faces (edges) of the octahedron. An edge (v_i, v_j) is described by the set of $2n - 1$ equations

$$x_i + x_j = 1$$

$$x_\ell = 0 \text{ for } \ell = 1, 2, \ldots, 2n$$

$$\ell \neq i \text{ or } j$$

The intersection of H with (v_i, v_j) is the point resulting from the addition of one of the equations

$$g_i x_i + g_j g_j = 0 \qquad (4.13)$$

$$g_i x_i - g_j x_j = 0 \qquad (4.14)$$

to the set describing (v_i, v_j), making 2n equations in all. Equation (4.13) appears if i and j are both either greater than n or less than n + 1;

Table 4.1

Sample Iteration of a Four-dimensional G-set

$\vec{I}^+$		$\vec{I}^-$	
x_1	x_2	x_3	x_4
.4	.6	.6	.2

$\vec{D}^+$		$\vec{D}^-$	
d_1	d_2	d_3	d_4
-.1	.1	.2	-.2

	G^+		G^-		
Iteration	g_1	g_2	g_3	g_4	R
1	.5	.5	.5	.5	.1*
2	.49	.51	.48	.52	.11
3	.48	.52	.46	.54	.12
4	.47	.53	.44	.56	.13
5	.45	.55	.41	.59	.15
6	.43	.57	.38	.62	.16
7	.42	.58	.35	.65	.18*
8	.40	.60	.31	.69	.20
9	.38	.62	.27	.73	.22
10	.36	.64	.23	.77	.24
11	.34	.66	.18	.82	.26
12	.31	.69	.13	.87	.29
13	.28	.72	.07	.93	.32*
14	.25	.75	.01	.99	.35
15	.21	.79	0.0	1.0	.36
⋮					
final	0.0	1.0	0.0	1.0	.40*

*Indicates points taken for example in Figure 4.3

Equation (4.14) appears otherwise. The former corresponds to the cases where x_i and x_j are both in I^+ or I^-, and the latter to the cases where x_i is in I^+ and x_j in I^- or x_i is in I^- and x_j is in I^+.

The solution to (4.12) + (4.13) is, for $g_j > 0$, $g_i > 0$

$$x_i = \frac{-g_j}{g_i - g_j} \quad , \quad x_j = \frac{g_i}{g_i - g_j} \tag{4.15}$$

This solution lies outside the principal section of the octahedron, since if x_i is positive, $g_i - g_j$ must be negative, and if $g_i - g_j$ is negative, x_j will be negative. Similarly, if $g_i - g_j$ is positive, x_i will be negative.

H and (v_i, v_j) can intersect only when $g_i = 0$, or $g_j = 0$, or both. When $g_i = 0$, the solution is $x_i = 1$, $x_j = 0$. When $g_j = 0$, the solution is $x_j = 1$, $x_i = 0$. These points correspond respectively to v_i and v_j. When $g_i = g_j = 0$, every point in (v_i, v_j) is a solution. Geometrically, the first case occurs when H passes through the vertex v_i, the second occurs when H passes through v_j, and the third when all of (v_i, v_j) lies in H.

The solution to (4.12) + (4.14) is

$$x_i = \frac{g_j}{g_i + g_j} \qquad x_j = \frac{g_i}{g_i + g_j} \tag{4.16}$$

which clearly lies in the principal section. The point (x_i, x_n) will lie closer to the vertex corresponding to the lesser of g_i, g_j, and will be at the midpoint of the edge (v_i, v_j) when $g_i = g_j$.

These solutions can be characterized by saying that H separates I^+ vertices from I^- vertices. In other words, if the input $\vec{x}$ lies on the side of the I^-, there is no response, and if $\vec{x}$ lies on the side of the I^+, there is a response. We will speak of H dividing the principal section into a response region and a non-response region.

An example is given in Figure 4.2 for the 4-dimensional case, as seen in the 4^{th} order face. H intersects the face in a plane H'. $I^+ = (x_1, x_2)$, $I^- = (x_3, x_4)$, $g_1 = .5$, $g_2 = .5$, $g_3 = .7$, $g_4 = .3$. Note that H intersects the edges (v_1, v_3), (v_1, v_4), (v_2, v_3) and (v_2, v_4); the edge (v_1, v_2) lies on the response side of H and the edge (v_3, v_4) lies on the nonresponse side of H.

The alteration of the G-sets results in a rotation of H about the line L. In the face this becomes a rotation about the center point. This alteration can proceed until the terminal state. Recall that the terminal state for the g-sets has each g_i equal to 0 or $\frac{1}{p}$. Taking every combination of g_i and g_j, we note the only possible combination of values are that both g_i and g_j are zero, that both are equal to $\frac{1}{p}$, and that one is equal to zero and the other equal to $\frac{1}{p}$. In the first instance, (v_i, v_j) lies entirely in H, in the second, H intersects (v_i, v_j) at its midpoint, and in the last, H intersects (v_i, v_j) at the vertex corresponding to the g-element equal to zero.

An example of the rotation process for $n = 2$ is given in Figure 4.3. The sequence of numbers is that given in Table 4.1. Note that the angle between H and (v_2, v_4) tends toward $90°$ and the angle between H and (v_1, v_3) tends toward $0°$.

The angle between H and an edge (v_i, v_j) is of interest since it tells how much the preponderance of x_i over x_j affects the output, which is related to the distance of $\vec{x}$ from H. As the projection of $\vec{x}$ on the $x_i x_j$ coordinate plane moves along the edge (v_i, v_j), the output will change as the cosine of the angle between $\vec{x}$ and $\vec{g}$, the normal to H, as in equation (2.2). If the angle between (v_i, v_j) with H becomes small, any movement of $\vec{x}$ primarily in the $x_i x_j$-coordinate plane will have little effect on the output, since this movement is perpendicular to $\vec{g}$. In other words it does not matter what value a particular x_i takes if the corresponding value of g_i is near zero.

Since the parameter p is the same for both G^+ and G^-, there will always be the same number of nonzero elements in both sets. The terminal case will, therefore, always have an even number of nonzero g_i and an even number of zero g_i. The set of (v_i, v_j) such that both g_i and g_j are zero determines a simplex of $2(n - p)$ dimensions. This simplex has as vertices all those coordinates which contribute nothing to the input. These vertices

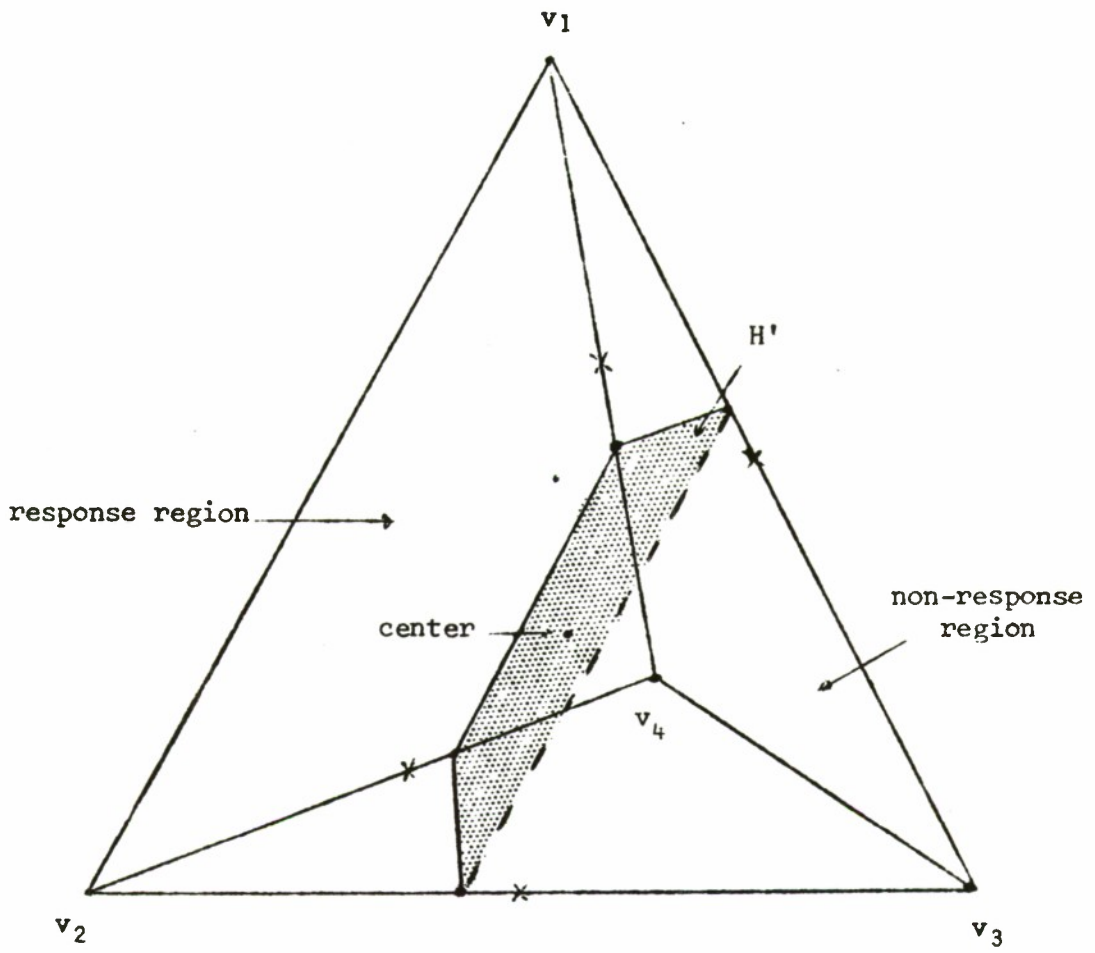


Figure 4.2

Four-dimensional Case for $g_1 = .5$, $g_2 = .5$, $g_3 = .7$, $g_4 = .3$

x indicates midpoint of edge.

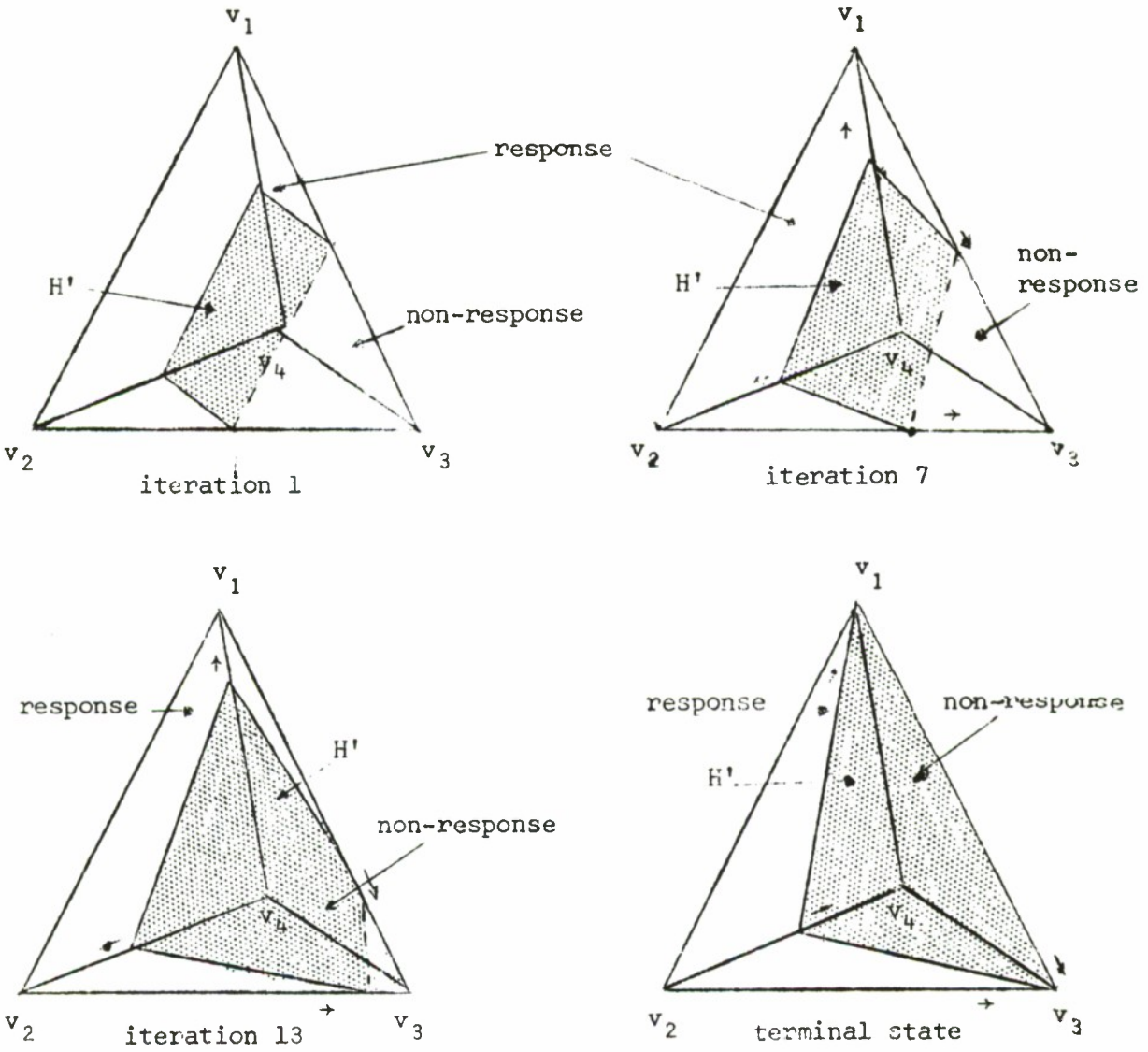


Figure 4.3 Various Stages in the Rotation of H' for Four Dimensions

Data taken from Table 1.

Arrows indicate direction of movement of intersections with edges.

can therefore be removed from the model, as in the discussion of the terminal state model in Equation (4.1) and following. We are then left with a $2p^{th}$ order simplex. In the example of Figure 4.3, the terminal simplex is the 2^{nd} order (v_2, v_4).

As discussed in Section III, the terminal simplex eventually reached is given by the p largest (smallest) elements of I^+ and the p smallest (largest) elements of I^-, depending on whether F is positive or negative. An example of inhibiting adaptation is given in Table 4.2. Note that the adaptation becomes slower as R decreases.

Since the terminal state is given by the indices of the p largest elements in D^+ and the p smallest elements in D^-, a large class of inputs will give rise to the same terminal state. This class is given by the region in the principal section determined by the set of $(2n - 1)$-dimensional hyperplanes described in detail in Appendix I. Each primary region in Appendix I corresponds to one of the possible total orderings of $\{x_1,\ldots,x_{2n}\}$. One of these primary regions can be characterized by the ordering $x_{i_1} > \cdots > x_{i_{2n}}$. The class of inputs which gives rise to the terminal simplex given by $(v_{i_1},\ldots,v_{i_p}, v_{i_{p+1}},\ldots,v_{i_{2p}})$ is a union of primary regions. Any ordering is included in the union in which all x_i in $\{x_{i_1},\ldots,x_{i_p}\}$ are greater than any other x_i in $\{x_1,\ldots,x_n\}$; and all x_i in $\{x_{i_{p+1}},\ldots,x_{i_{2k}}\}$ are less than any other x_i in $\{x_{n+1},\ldots,x_{2n}\}$.

An example will clarify this concept. In the Figure 5 of Appendix I, a 4-dimensional case, there are 24 primary regions characterized by the orderings

$$
\begin{aligned}
&x_1 > x_2 > x_3 > x_4\\
&x_1 > x_2 > x_4 > x_3\\
&x_1 > x_3 > x_2 > x_4\\
&x_1 > x_3 > x_4 > x_2\\
&\qquad\vdots\\
&x_4 > x_3 > x_2 > x_1 \quad .
\end{aligned}
$$

The terminal simplex is (v_2, v_4), so that the regions:

Table 4.2

Inhibiting Rotation of Example of Table 4.1 (F = -1)

$\vec{I}^+$	
x_1	x_2
.4	.6

$\vec{I}^-$	
x_3	x_4
.6	.2

$\vec{D}^+$	
d_1	d_2
-.1	.1

$\vec{D}^-$	
d_3	d_4
.2	-.2

	$\vec{G}^+$		$\vec{G}^-$		R
Iteration	g_1	g_2	g_3	g_4	
1	.5	.5	.5	.5	.1
2	.51	.49	.52	.48	.09
3	.52	.48	.54	.46	.08
4	.53	.47	.55	.45	.07
5	.53	.47	.56	.44	.07
6	.54	.46	.58	.42	.06
7	.54	.46	.59	.41	.06
8	.55	.45	.60	.40	.05
9	.55	.45	.61	.39	.05
10	.56	.44	.62	.38	.04
11	.56	.44	.63	.37	.036
12	.556	.444	.637	.363	.034
13	.553	.447	.644	.356	.032

$$
\begin{aligned}
&x_2 > x_1 > x_3 > x_4 \\
&x_2 > x_3 > x_1 > x_4 \\
&x_2 > x_3 > x_4 > x_1 \\
&x_3 > x_2 > x_1 > x_4 \\
&x_3 > x_2 > x_4 > x_1 \\
&x_3 > x_4 > x_2 > x_1
\end{aligned}
\tag{4.18}
$$

together contain all inputs which give rise to the terminal simplex of Figure 4.3.

When there is a sequence of different inputs to the neuromime, the rotation will be in the same direction for each one provided that the inputs all lie in the same region. If some of the inputs lie in a different region, there will be competition between the two rotations, and the attainment of the terminal state will be delayed. The state which will predominate is not immediately clear, since the order of inputs in the sequence is quite significant.

These unions of primary regions are the sets of inputs which the neuromime considers similar in a certain sense, so that they indicate which classes of inputs may most easily be identified by the neuromime. As seen above, an attempt to force the neuromime to recognize a set of inputs lying in different regions will result in varying results and slower adaptation.

At this point, it is useful to make a final examination of the vector aspects of the computation of R, especially the algebraic aspects. The definitions of the vectors involved are recapitulated below:

$$\vec{x} = \begin{pmatrix} x_1^+ \\ \vdots \\ x_n^+ \\ x^- \\ \vdots \\ x_n^- \end{pmatrix} \qquad \vec{m} = \begin{pmatrix} \bar{x}^+ \\ \vdots \\ \bar{x}^+ \\ \bar{x}^- \\ \vdots \\ \bar{x}^- \end{pmatrix} \tag{4.19}$$

$$\vec{d} = \begin{pmatrix} d_1^+ \\ \vdots \\ d_n^+ \\ d_1^- \\ \vdots \\ d_n^- \end{pmatrix} \qquad \vec{g} = \begin{pmatrix} g_1 \\ \vdots \\ g_n \\ -g_{n+1} \\ \vdots \\ -g_{2n} \end{pmatrix}$$

we will need the identities

$$\vec{x} = \vec{d} + \vec{m} \tag{4.20}$$

$$\vec{x} \cdot \vec{d} = (\vec{d} + \vec{m}) \cdot \vec{d} = |\vec{d}|^2 + \vec{d} \cdot \vec{m} \tag{4.21}$$

$$\vec{d} \cdot \vec{m} = \bar{x}^+ \sum_{i=1}^{n} d_i^+ + \bar{x}^- \sum_{i=1}^{n} d_i^- = 0 \tag{4.22}$$

so that

$$\vec{x} \cdot \vec{d} = |\vec{d}|^2 \tag{4.23}$$

The space of possible $\vec{x}$ vectors is a vector space of $2n$ dimensions, denoted by X_{2n}. Each $\vec{x} \in X_{2n}$ can be written as $\vec{d} + \vec{m}$, and, from (4.22) $\vec{d} \cdot \vec{m} = 0$. The components of each vector $\vec{m}$ satisfy the constraints

$$m_1 = \cdots = m_n, \; m_{n+1} = \cdots = m_{2n} \tag{4.24}$$

The vectors $\vec{m}$ therefore lie in a space M_2 of two dimensions. M_2 contains the zero vector. Since each component of each $\vec{d}$ satisfies the constraints

$$d_1 + \cdots + d_n = d_{n+1} + \cdots + d_{2n} = 0 \ , \tag{4.25}$$

The only $\vec{d}$ that is contained in M_2 is the zero vector. X_{2n} can therefore be decomposed into

$$X_{2n} = M_2 \oplus D_{2n-2} \ , \tag{4.26}$$

with every $\vec{d}$ contained in D_{2n-2}, where $\oplus$ denotes an orthogonal direct sum as in Hoffman and Kunze (1961).

Considered as a hyperplane in the principal section of the octahedron, M_2 is a plane passing through the origin, the center of the face, and the centers of both the simplexes P and N given respectively by

$$x_1 + \cdots + x_n = a \ . \ x_{n+1} = \cdots = x_{2n} = 0 \tag{4.27}$$

and

$$x_1 = \cdots = x_n = 0 \ , \ x_{n+1} + \cdots + x_{2n} = a \tag{4.28}$$

$$0 \leq a \leq 1$$

In other words, M_2 is the direct sum of the orthogonal lines Q, Q' given respectively by

$$x_1 = \cdots = x_{2n} \tag{4.29}$$

and

$$x_1 = \cdots = x_n \ , \ x_{n+1} = \cdots = x_{2n} \ , \ x_1 = -x_{n+1} \tag{4.30}$$

D_{2n-2} therefore is a hyperplane perpendicular to M_2--it corresponds to the set of hyperplanes parallel to H_0, where all the g-elements are equal. The two components of M_2 in Q and Q' can be thought of as determining

respectively the distance from the origin to the face to be considered and the distance from the center of the face to the hyperplane parallel to H_0 containing $\vec{x}$.

Define $\vec{f}$ and $\vec{\ell}$ to be the components of $\vec{m}$ in Q and Q' respectively. Note that

$$\vec{g} = \begin{pmatrix} g_1 \\ \vdots \\ g_n \\ -g_{n+1} \\ \vdots \\ -g_{2n} \end{pmatrix} \tag{4.31}$$

is the vector normal to H , and is not, in general, normal to H_0 . From (4.29) and (4.31)

$$\vec{f} \cdot \vec{g} = 0 \tag{4.32}$$

From (4.32) and (4.26)

$$\vec{g} \cdot \vec{x} = \vec{g} \cdot (\vec{f} + \vec{\ell} + \vec{d}) = \vec{g} \cdot \vec{\ell} + \vec{g} \cdot \vec{d} \tag{4.33}$$

and if, in addition, $\vec{g} = \vec{g}_0$, noting (4.30),

$$\vec{g}_0 \cdot \vec{x} = \vec{g}_0 \cdot \vec{\ell} = a \ . \tag{4.34}$$

From (4.30), a is the difference between the average of the x_i^+ and the average of the x_i^-; and the output of a neuromime is never less than 0; so that when $a \leq 0$, we have $R = 0$.

This implies that the significant parameter in determining the initial strength of an input is the difference between the average of the I^+ and the average of the I^- . In tables 4.1 and 4.2, we see that following (4.34),

$$\begin{aligned} \bar{x}^+ &= .5 \\ \bar{x}^- &= .4 \\ a &= .1 \\ R(0) &= a = .1 \end{aligned} \tag{4.35}$$

The terminal output for positive adaptation is given by the difference between the average of the P largest in I^+ and p smallest in I^-. In Table 4.1, this average is .4, so that

$$R_{final} = .4 \tag{4.36}$$

and R will go no higher.

For negative adaptation, the terminal state will be reached only if the average of the p smallest in I^+ is greater than or equal to the average of the p largest in I^-, since otherwise R will go to zero before the terminal state is reached. In the example of Table 4.2, this difference is -.2. As can be seen, the g-sets are nowhere near a terminal state, yet the output is very near zero. The adaptation equations developed in Section 5 can be solved to get the g-set for which R becomes zero. These are solved in Appendix II.

It remains to consider the effect of the bias and the S inputs on this geometrical model. These additional parameters can be regarded from Equation (2.1) as constituting an additive constant to the measurement Equation (4.1). Equation (2.1) becomes

$$\begin{aligned} c &= B + S \\ R &= \vec{x} \cdot \vec{g} + c \,. \end{aligned} \tag{4.37}$$

Expressing $\vec{x} \cdot \vec{g}$ as $|\vec{x}|\ |\vec{g}| \cos\theta$, setting $R = 0$, and solving (4.37) for θ, we have

$$|\vec{x}|\ |\vec{g}|\ \cos\theta = -c$$

$$\theta = \cos^{-1}\left(-\frac{c}{|\vec{x}|\ |\vec{g}|}\right) \tag{4.38}$$

Angle θ is the angle between the $\vec{x}$ vector and the fixed $\vec{g}$ vector which determines the locus of points for which $R = 0$. Any $\vec{x}$ whose angle with $\vec{g}$ is greater than θ will not produce an output. Note that the value of this critical angle depends on the magnitude of $\vec{x}$. Figure 4.4 shows a sketch of θ versus $|\vec{x}|$.

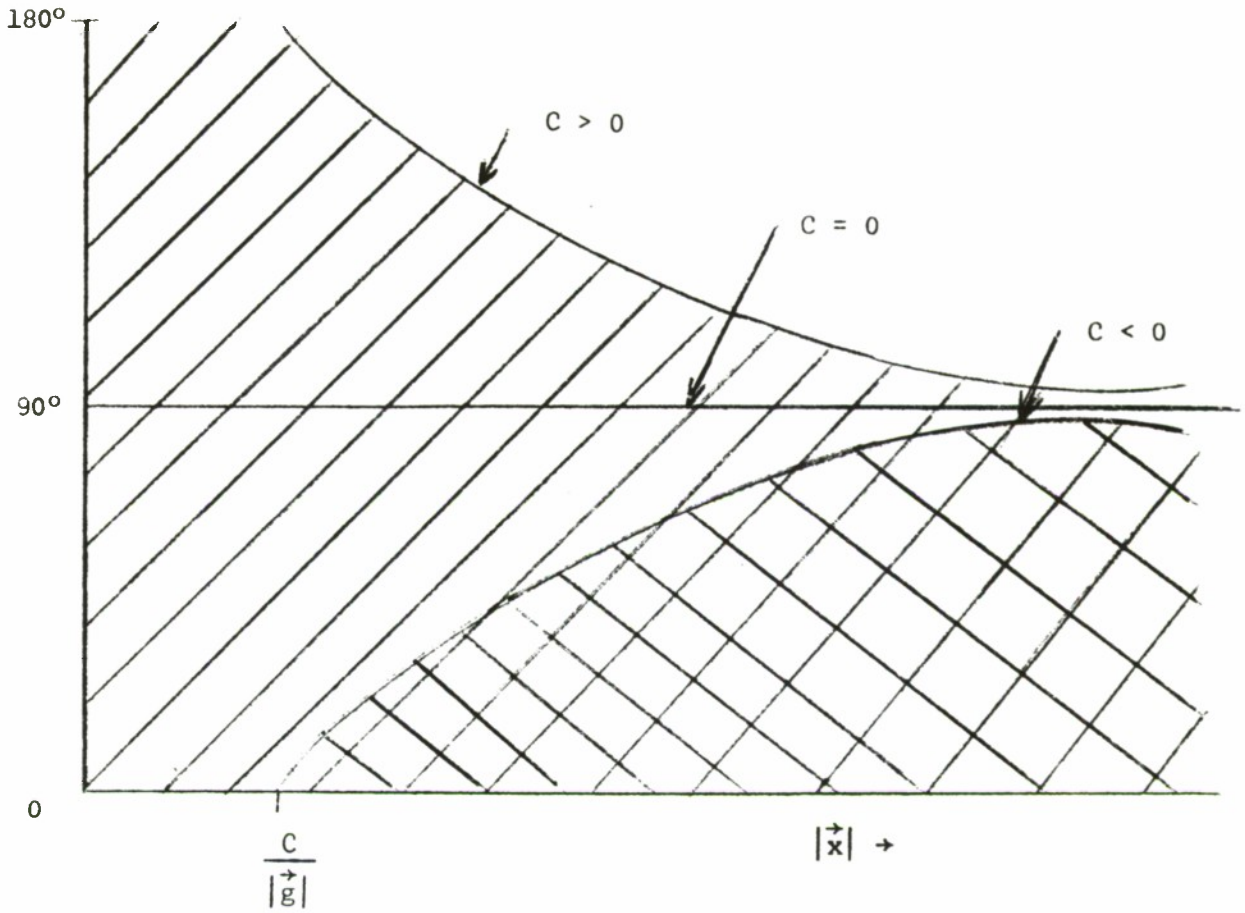


Figure 4.4
Sketch of Critical Angle

$$\theta = \cos^{-1}\left(-\frac{c}{|\vec{g}|\ |\vec{x}|}\right)$$

for $c > 0$, $c = 0$, $c < 0$

Justification:

$$\frac{d\theta}{dx} = \frac{1}{\sqrt{1 - \frac{c^2}{|\vec{x}|^2\ |\vec{g}|^2}}} \to 0 \text{ as } |\vec{x}| \to \infty$$

For $c > 0$, $\theta = 180°$ when $|\vec{x}| = \frac{c}{|\vec{g}|}$

For $c < 0$ $\theta = 0°$ when $|\vec{x}| = \frac{c}{|\vec{g}|}$

As might be expected, for $c = 0$, $\theta = 0$ for any value of $|\vec{x}|$. This corresponds to the hyperplane H of (4.11) which is the locus of $\vec{x}$ orthogonal to $\vec{g}$. When $c < 0$, no $\vec{x}$ will produce an output if its magnitude is less than $c/|\vec{g}|$, since no matter how close to $\vec{g}$ it is, the output cannot overcome the negative bias. Conversely, when $c > 0$, every $\vec{x}$ whose magnitude is less than $c/|\vec{g}|$ will produce an output, since no matter how far from $\vec{g}$ it is, it cannot overcome the positive bias. In both cases, as $|\vec{x}|$ increases, the effect of the bias is reduced in that as θ approaches $90°$, the critical value approaches that for $c = 0$.

The effect of the bias is to change the volume of the response area. When $c < 0$, the volume is decreased, and when $c > 0$, it is increased. It is somewhat difficult to visualize this phenomenon in the octahedron model, but the hatched areas under the curves of Figure 4.4 give some idea of how it works. For $c > 0$, the response area is greater than for $c = 0$, and for $c > 0$, the response area is less.

When the bias is large, the threshold hyperplane can be rotated by the adaptation process so that when the bias is again reduced the components of the level will give maximal response to new inputs. This property can be used to cause neuromimes near terminal states to "forget" their conditioning so that they can be used for new tasks.

As described in Section II, the S inputs are interaction inputs between different neuromimes on the same level. There is an adaption process for these inputs as well as for the I inputs, which was described in Section II when there is only one neighbor connected to both an S^+ and an S^- input. There can be a large number of such connections, and the S g-sets behave much like the I g-sets. It makes more sense, however, to consider the S inputs in a model as in Section III and take the effect on the geometrical model as one of additional bias.

The adaptation process on a large number of S inputs will cause the degree of interaction to alter differentially among them, so that the influence of those neighbors with high outputs will be altered differently from that of those with small outputs. As with the G_I sets, the controlling factors in the differentiation are the dispersion among the S inputs and the maximum allowable value for an individual g_i.

The equations involved are:

$x_i^{+,-}$ are the individual components of $\vec{S}^{+,-}$

$$\vec{D}_S^{+,-} = \vec{S}^{+,-} - \vec{\bar{S}}^{+,-}$$

$d_i^{+,-}$ are the individual components of $\vec{D}_S^{+,-}$

$$\vec{G}_S^{+,-} = \vec{G}_S^{+,-} + F_S \cdot R \cdot \vec{D}_S^{+,-}$$

$g_i^{+,-}$ are the individual components of $\vec{G}_S^{+,-}$

p_S is the number of g_i^+ and g_i^- allowed to reach a nonzero terminal state.

When F_S is positive, those g_i^+ corresponding to positive d_i^+ will be increased and those corresponding to negative d_i^+ will be decreased, while those g_i^- corresponding to positive d_i^- will be decreased and those g_i^- corresponding to negative d_i^- will be increased. The terminal state is that the g_i^+ corresponding to the p_S highest d_i^+ and the g_i^- corresponding to the p_S lowest d_i^- will be equal to $1/p_S$, while all other $g_i^{+,-}$ will be equal to zero. When F_S is negative, the reverse occurs.

In this manner, the degree of interaction among a component and its neighbors can be limited to the most significant (F_S positive) or to the least significant (F_S negative) for a particular class of stimuli.

It should be noted that the outputs of a level form the same sort of signal space as the inputs, so that the outputs of one level can be used as inputs to other levels.

Section V

ADAPTATION IN A SINGLE NEUROMIME

A question of some importance in the operation of a neuromime is the way in which the adaption process takes place and, in particular, how the various parameters affect the adaptation. We will again deal with the problem with $M_S = 0$, but will allow a non-zero bias B. Later in the section, the general case will be considered. We will assume throughout piecewise constant inputs.

The adaption equations, given in Equations (2.2) in incremental form can be expressed in differential form, using the notation of Section III.

$$\begin{aligned} \frac{d}{dt}\vec{G}^{+} &= F\vec{RD}^{+} \\ \frac{d}{dt}\vec{G}^{-} &= -F\vec{RD}^{-} \end{aligned} \tag{5.1}$$

where, using the notation of (4.9)

$$R = M_I\left(\sum_{i=1}^{n} g_i x_i - \sum_{i=n+1}^{2n} g_i x_i\right) + B$$

The Equations (5.1) may be expressed as the system

$$\begin{aligned}
\frac{d}{dt} g_1^{+} &= FM_I(d_1^{+}x_1^{+}g_1^{+} + \cdots + d_1^{+}x_n^{+}g_n^{+}) + FM_I(-d_1^{+}x_1^{-}g_1^{-} - \cdots - d_1^{+}x_n^{-}g_n^{-}) + FBd_1^{+} \\
&\vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\
\frac{d}{dt} g_n^{+} &= FM_I(d_n^{+}x_1^{+}g_1^{+} + \cdots + d_n^{+}x_n^{+}g_n^{+}) + FM_I(-d_n^{+}x_1^{-}g_1^{-} - \cdots - d_n^{+}x_n^{-}g_n^{-}) + FBd_n^{+} \\
-\frac{d}{dt} g_1^{-} &= FM_I(d_1^{-}x_1^{+}g_1^{+} + \cdots + d_1^{-}x_n^{+}g_n^{+}) + FM_I(-d_1^{-}x_1^{-}g_1^{-} - \cdots - d_1^{-}x_n^{-}g_n^{-}) + FBd_1^{-} \\
&\vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\
-\frac{d}{dt} g_n^{-} &= FM_I(d_n^{-}x_1^{+}g_1^{+} + \cdots + d_n^{-}x_n^{+}g_n^{+}) + FM_I(-d_n^{-}x_1^{-}g_1^{-} - \cdots - d_n^{-}x_n^{-}g_n^{-}) + FBd_n^{-}
\end{aligned} \tag{5.2}$$

Adopting the definitions, as in Equation (4.10)

$$\vec{g} = \begin{pmatrix} g_1^+ \\ \vdots \\ g_n^+ \\ -g_1^- \\ \vdots \\ -g_n^- \end{pmatrix} \quad \vec{u} = \begin{pmatrix} d_1^+ \\ \vdots \\ d_n^+ \\ d_1^- \\ \vdots \\ d_n^- \end{pmatrix} \quad \vec{v} = \begin{pmatrix} FM_I x_1^+ \\ \vdots \\ FM_I x_n^+ \\ FM_I x_1^- \\ \vdots \\ FM_I x_n^- \end{pmatrix} = FM_I \vec{x}$$

The system (5.2) may be re-expressed in the more compact form

$$\frac{d}{dt}\vec{g} = (\vec{u} \odot \vec{v}^t) \odot \vec{g} + FB\vec{u} \tag{5.3}$$

where $\vec{v}^t$ is the transpose of $\vec{v}$ (i.e., the row matrix representation) and the operation $\odot$ represents matrix multiplication.

In Appendix III the solution to the system (5.3) is given by

$$\vec{g} = \vec{c}ue^{\lambda t} + \vec{k} - \frac{FB}{\lambda}\vec{u} \tag{5.4}$$

where c is an arbitrary constant, and $\vec{k}$ is a vector of $2n$ arbitrary constants the last element of which is determined, and

$$\begin{aligned} \lambda = \vec{u} \cdot \vec{v} &= FM_I\left[\sum_{i=1}^{n} d_i^+ x_i^+ + \sum_{i=1}^{n} d_i^- x_i^-\right] \\ &= FM_I\left[\sum_{i=1}^{n} d_i^+ (d_i^+ + \bar{x}^+) + \sum_{i=1}^{n} d_i^- (d_i^- + \bar{x}^-)\right] \\ &= FM_I\left[\sum_{i=1}^{n} d_i^{+2} + \sum_{i=1}^{n} d_i^{-2} + \bar{x}^+ \sum_{i=1}^{n} d_i^+ + \bar{x}^- \sum_{i=1}^{n} d_i^-\right] \\ &= FM_I\left[\sum_{i=1}^{n} d_i^+ + \sum_{i=1}^{n} d_i^{-2}\right] \end{aligned} \tag{5.5}$$

since from Equation (2.5),

$$\sum_{i=1}^{n} d_i^+ = \sum_{i=1}^{n} d_i^- = 0$$

It will be convenient to simplify the notation, removing the superscript + and - , and expressing the g_i as functions of the time t , as follows:

$$g_i(t) = g_i^+$$

$$g_{n+i}(t) = g_i^-$$

$$d_i = d_i^+ \qquad \text{for } i = 1, \ldots, n$$

$$d_{n+i} = d_i^-$$

so that (5.5) is expressed as

$$\lambda = FM_I \sum_{i=1}^{2n} d_i^2 \tag{5.6}$$

The summation in (5.6) is important in the following, and will be referred to by the symbol

$$\delta = \sum_{i=1}^{2n} d_i^2 \tag{5.7}$$

so that

$$\lambda = FM_I\delta \tag{5.8}$$

The system (5.4) becomes

$$\begin{aligned}
g_1(t) &= cd_1 e^{FM_I\delta t} + k_1 - \frac{B}{M_I\delta} d_1 \\
&\vdots \quad\quad \vdots \quad\quad \vdots \quad\quad \vdots \\
g_n(t) &= cd_n e^{FM_I\delta t} + k_n - \frac{B}{M_I\delta} d_n \\
g_{n+1}(t) &= -cd_{n+1} e^{FM_I\delta t} - k_{n+1} + \frac{B}{M_I\delta} d_{n+1} \\
&\vdots \quad\quad \vdots \quad\quad \vdots \quad\quad \vdots \\
g_{2n}(t) &= -cd_{2n} e^{FM_I\delta t} - k_{2n} + \frac{B}{M_I\delta} d_{2n}
\end{aligned} \tag{5.9}$$

System (5.9) must be solved for the constants c and $k_1, \ldots, k_n$ by evaluating for $t = 0$ and the initial values of the $g_i(t)$,

$$\begin{aligned}
g_1(0) &= cd_1 + k_1 - \frac{B}{M_I\delta} d_1 \\
&\vdots \quad \vdots \quad \vdots \quad \vdots \\
g_n(0) &= cd_n + k_n - \frac{B}{M_I\delta} d_n \\
g_{n+1}(0) &= -cd_{n+1} - k_{n+1} + \frac{B}{M_I\delta} d_{n+1} \\
&\vdots \quad \vdots \quad \vdots \quad \vdots \\
g_{2n}(0) &= -cd_{2n} - k_{2n} + \frac{B}{M_I\delta} d_{2n}
\end{aligned} \tag{5.10}$$

with the addition of the constraint on k_{2n} from Appendix III,

$$\begin{aligned}
k_{2n} &= \frac{FM_1x_1k_1 + \cdots + FM_Ix_nk_n + FM_Ix_{n+1}k_{n+1} + \cdots + FM_Ix_{2n-1}k_{2n-1}}{-FM_Ix_{2n}} \\
&= \frac{x_1k_1 + \cdots + x_nk_n + x_{n+1}k_{n+1} + \cdots + x_{2n-1}k_{2n-1}}{-x_{2n}}
\end{aligned} \tag{5.11}$$

Solving (5.10), we obtain

$$c = \frac{g_1(0) - k_1 + \frac{B}{M_I\delta} d_1}{d_1} \tag{5.12}$$

$$k_i = g_i(0) - cd_i + \frac{B}{M_I\delta} d_i \qquad i = 1, \ldots, n$$

$$k_i = -g_i(0) - cd_i + \frac{B}{M_I\delta} d_i \qquad i = n+1, \ldots, 2n$$

k_1 can be obtained by equating the expressions for k_{2n} in (5.11) and (5.12), substituting the expression (5.12) for k_i in (5.11) with $i = 2, \ldots, 2n-1$; and substituting for c the expression in (5.12).

$$\frac{x_1 k_1}{-x_{2n}} + \sum_{i=2}^{n} \frac{x_i}{-x_{2n}} \left[g_i(0) - \frac{\left(g_1(0) - k_1 + \frac{B}{M_I \delta} d_1 \right)}{d_1} d_i + \frac{B}{M_I \delta} d_1 \right]$$

$$+ \sum_{i=n+1}^{2n-1} \frac{x_i}{-x_{2n}} \left[-g_i(0) - \frac{\left(g_1(0) - k_1 + \frac{B}{M_I \delta} d_1 \right)}{d_1} d_i + \frac{B}{M_I \delta} d_i \right]$$

$$= -g_{2n}(0) - \frac{\left(g_1(0) - k_1 + \frac{B}{M_I \delta} d_1 \right)}{d_1} d_{2n} + \frac{B}{M_I \delta} d_{2n} \qquad (5.13)$$

Multiplying both sides of (5.13) by $-d_1 x_{2n}$ and expanding,

$$x_1 k_1 d_1 + \sum_{i=2}^{n} \left[x_i g_i(0) d_1 - x_i g_1(0) d_i + x_i k_1 d_i - \frac{B}{M_I \delta} d_1 x_i d_i + \frac{B}{M_I \delta} d_1 x_i d_i \right]$$

$$+ \sum_{i=n+1}^{2n-1} \left[-x_i g_i(0) d_1 - x_i g_1(0) d_i + x_i k_1 d_i - \frac{B}{M_I \delta} d_1 x_i d_i + \frac{B}{M_I \delta} d_1 x_i d_i \right] \qquad (5.14)$$

$$= g_{2n}(0) x_{2n} d_1 + x_{2n} g_1(0) d_{2n} - x_{2n} k_1 d_{2n} + \frac{B}{M_I \delta} x_{2n} d_{2n} d_1 - \frac{B}{M_I \delta} x_{2n} d_{2n} d_1$$

Gathering terms in (5.14) noting that all terms containing $\frac{B}{M_I \delta}$ drop out, we obtain

$$k_1 \left[x_1 d_1 + \sum_{i=2}^{n} x_i d_i + \sum_{i=n+1}^{2n-1} x_i d_i + x_{2n} d_{2n} \right] =$$

$$d_1 x_{2n} g_{2n}(0) + g_1(0)\, x_{2n} d_{2n} - \sum_{i=2}^{n} d_1 x_i g_i(0) + \sum_{i=2}^{n} g_1(0)\, x_i d_i \qquad (5.15)$$

$$+ \sum_{i=n+1}^{2n-1} d_1 x_i g_i(0) + \sum_{i=n+1}^{2n-1} g_1(0)\, x_i d_i$$

The left side of (5.15) can be simplified,

$$k_1\left[x_1d_1 + \sum_{i=2}^{n} x_id_i + \sum_{i=n+1}^{2n-1} x_id_i + x_{2n}d_{2n}\right] = k_1 \sum_{i=1}^{2n} x_id_i = k_1 \sum_{i-1}^{2n} d_i^2 = k_1\delta \quad . \tag{5.16}$$

The right side of (5.15) can also be simplified, by adding and subtracting the term $g_1(0)x_1d_1$ and rearranging,

$$g_1(0)\, x_1d_1 + g_1(0) \sum_{i=2}^{n} x_id_i + g_1(0) \sum_{i=n+1}^{2n-1} x_id_i + g_1(0)\, x_{2n}d_{2n}$$

$$-d_1x_1g_1(0) - d_1 \sum_{i=2}^{n} x_ig_i(0) + d_1 \sum_{i=n+1}^{2n-1} x_ig_i(0) + d_1x_{2n}g_{2n}(0)$$

$$= g_1(0) \sum_{i=1}^{2n} x_id_i - d_1\left[\sum_{i=1}^{n} x_ig_i(0) - \sum_{i=n+1}^{2n} x_ig_i(0)\right] \quad . \tag{5.17}$$

Defining, according to (5.1),

$$R(0) = M_I\left[\sum_{i=1}^{n} x_ig_i(0) - \sum_{i=n+1}^{2n} x_ig_i(0)\right] + B \tag{5.18}$$

and substituting (5.16) and (5.17), equation (5.15) becomes

$$k_1\delta = g_1(0)\delta - d_1\left[\frac{R(0) - B}{M_I}\right] \tag{5.19}$$

or

$$k_1 = g_1(0) - \frac{d_1}{\delta}\left[\frac{R(0) - B}{M_I}\right] \quad . \tag{5.20}$$

Substituting (5.20) into the expression in (5.12) for c,

$$c = \frac{g_1(0) - g_1(0) + \frac{d_1}{\delta}\left[\frac{R(0) - B}{M_I}\right] + \frac{B}{M_I\delta}\, d_1}{d_1} \tag{5.21}$$

or, simplifying,

$$c = \frac{1}{M_I\delta}\left[R(0) - B + B\right] = \frac{R(0)}{M_I\delta} \tag{5.22}$$

Substituting the expression for the k_i from (5.12) into the system (5.9) we obtain the solution to (5.1),

$$\begin{aligned}
g_1(t) &= cd_1 e^{FM_I\delta t} + g_1(0) - cd_1 \\
&\vdots \quad \vdots \quad \vdots \quad \vdots \\
g_n(t) &= cd_n e^{FM_I\delta t} + g_n(0) - cd_n \\
-g_{n+1}(t) &= cd_{n+1} e^{FM_I\delta t} - g_{n+1}(0) - cd_{n+1} \\
&\vdots \quad \vdots \quad \vdots \quad \vdots \\
-g_{2n}(t) &= cd_{2n} e^{FM_I\delta t} - g_{2n}(0) - cd_{2n}
\end{aligned} \tag{5.23}$$

In a vector notation, using the definitions in (5.1), (5.23) is

$$\begin{aligned}
\vec{G}^+(t) &= c\vec{D}^+ e^{FM_I\delta t} + \vec{G}^+(0) - c\vec{D}^+ \\
-\vec{G}^-(t) &= c\vec{D}^- e^{FM_I\delta t} - \vec{G}^-(0) - c\vec{D}^-
\end{aligned} \tag{5.24}$$

and in the form of the system (5.3); (5.23) is, substituting (5.22) for c,

$$\vec{g}(t) = \frac{R(0)}{M_I\delta}\vec{u}e^{FM_I\delta t} + \vec{g}(0) - \frac{R(0)}{M_I\delta}\vec{u} \tag{5.25}$$

or,

$$\vec{g}(t) = \frac{R(0)}{M_I\delta}\vec{u}\,(e^{FM_I\delta t} - 1) + \vec{g}(0) \tag{5.26}$$

Note that there are two basic controlling factors in (5.26), an exponential term and a constant multiplier term. The exponential term

$$\lambda = F_I M_I \delta \tag{5.27}$$

is controlled by the variance of the input and both the input level weight and the learning factor. The constant term

$$c = \frac{R(0)}{M_I \delta} = \frac{M_I \vec{x} \cdot \vec{g}(0) + B}{M_I \delta}$$

$$= \frac{\vec{x} \cdot \vec{g}(0)}{\delta} + \frac{B}{M_I \delta} \tag{5.28}$$

is dependent mostly on the bias term, since $\vec{x} \cdot \vec{g}_0$ can be of the same order of magnitude as δ.

We are now in a better position to attack the general problem with $M_S \neq 0$. There are several possible ways.

One important consideration is that we are interested in interconnections of the type used in Section II so that we want to determine the adaptation of the m-element vector

$$\vec{K} = \vec{G}_S^+ - \vec{G}_S^- \quad . \tag{5.29}$$

This can be determined by differentiating (5.29)

$$\frac{d}{dt}\vec{K} = \frac{d}{dt}\vec{G}_S^+ - \frac{d}{dt}\vec{G}_S^-$$

$$= F_S R(\vec{S}^+ - \vec{\bar{S}}^+) - [-F_S R(\vec{S}^- - \vec{\bar{S}}^-)] \tag{5.30}$$

If

$$\vec{S}^+ = \vec{S}^- = \vec{S}$$

and s_i is the i^{th} element of $\vec{S}$.

Equation (5.30) reduces to

$$\frac{d}{dt}\vec{K}(t) = 2F_S R(\vec{S} - \vec{\bar{S}}) \tag{5.31}$$

The simplest way of solving (5.31) is to consider the solution of the system for $M_S = 0$, then solve (5.31) with R held constant and the feedback delay of the net large enough so that the value of S remains constant. The solution is obtained immediately. Define

$$\vec{E} = \vec{S} - \vec{\bar{S}} \quad . \tag{5.32}$$

Then

$$\vec{K}(t) = 2F_S R\vec{E}t \quad . \tag{5.33}$$

A more realistic solution is to assume that R varies but that S remains constant. In this case, following the same arguments as in (5.1), the solution is

$$\vec{K}(t) = \frac{R(0)}{M_S \delta_2} \vec{E}(e^{\lambda t} - 1) + \vec{K}(0) \tag{5.34}$$

where

$$\lambda = 2F_S M_S \delta_2 \tag{5.35}$$

$$\delta_2 = \sum_{i=1}^{m} e_i^2 \tag{5.36}$$

and

e_i is the i^{th} component of $\vec{E}$.

The interactions between the two adaptations can be examined by solving the two systems simultaneously. The complete system, (2.2), using the notation of this section, is given by

$$\frac{d}{dt} g_i(t) = F_I d_i \left\{ M_I \left[\sum_{j=1}^{n} x_j g_j(t) - \sum_{j=n+1}^{2n} x_j g_j(t) \right] + B + M_S \sum_{j=1}^{m} s_j k_j(t) \right\}$$

$$\text{for } i = 1,\ldots,n$$

$$-\frac{d}{dt} g_i(t) = F_I d_i \left\{ M_I \left[\sum_{j=1}^{n} x_j g_j(t) - \sum_{j=n+1}^{2n} x_j g_j(t) \right] + B + M_S \sum_{j=1}^{m} s_j k_j(t) \right\}$$

$$\text{for } i = n+1,\ldots,2n$$

$$\frac{d}{dt} k_i(t) = 2F_S e_i \left\{ M_I \left[\sum_{j=1}^{n} x_j g_j(t) - \sum_{j=n+1}^{2n} x_j g_j(t) \right] + B + M_S \sum_{j=1}^{m} s_j k_j(t) \right\} \tag{5.37}$$

$$\text{for } i = 1,\ldots,m.$$

Following (5.3) system (5.37) can be expressed as

$$\frac{d}{dt} \vec{y}(t) = \vec{u} \odot \vec{v}^t \odot \vec{y}(t) + b\vec{u} \tag{5.38}$$

where

$$\vec{y}(t) = \begin{pmatrix} g_1(t) \\ \vdots \\ g_n(t) \\ -g_{n+1}(t) \\ \vdots \\ -g_{2n}(t) \\ k_1(t) \\ \vdots \\ k_m(t) \end{pmatrix} \quad \vec{u} = \begin{pmatrix} F_I d_1 \\ \vdots \\ F_I d_n \\ F_I d_{n+1} \\ \vdots \\ F_I d_{2n} \\ 2F_S e_1 \\ \vdots \\ 2F_S e_m \end{pmatrix} \quad \vec{v} = \begin{pmatrix} M_I x_1 \\ \vdots \\ M_I x_n \\ M_I x_{n+1} \\ \vdots \\ M_I x_{2n} \\ M_S s_1 \\ \vdots \\ M_S s_m \end{pmatrix}$$

and

$$b = B$$

so that the solution is given by Appendix III as

$$\vec{y}(t) = c\vec{u}e^{\lambda t} + \vec{z} - \frac{B}{\lambda}\vec{u} \tag{5.39}$$

where

$$\lambda = \vec{u} \cdot \vec{v}$$

$$= F_I M_I \sum_{i=1}^{2n} d_i^2 + 2F_S M_S \sum_{i=1}^{m} e_i^2 \quad . \tag{5.40}$$

Letting

$$\delta_1 = \sum_{i=1}^{2n} d_i^2 \ , \ \delta_2 = \sum_{i=1}^{m} e_i^2 \ ,$$

$$\lambda = F_I M_I \delta_1 + 2F_S M_S \delta_2 \quad . \tag{5.41}$$

Following the argument from (5.9) through (5.26),

$$c = \frac{R(0)}{\lambda} \tag{5.42}$$

where

$$R(0) = \vec{y} \cdot \vec{x} + B \tag{5.43}$$

and the solution is

$$\vec{y}(t) = \frac{R(0)}{\lambda} \vec{u}(e^{\lambda t} - 1) + \vec{y}(0) \quad . \tag{5.44}$$

The interaction between the two systems can be seen by examining in detail the solutions for $g_1(t)$ and $k_1(t)$.

$$g_1(t) = \frac{F_I R(0) d_1 (e^{\lambda t} - 1)}{F_I M_I \delta_1 + 2F_S M_S \delta_2} + g_1(0) \tag{5.45}$$

$$k_1(t) = \frac{2F_S R(0) e_1 (e^{\lambda t} - 1)}{F_I M_I \delta_1 + 2F_S M_S \delta_2} + k_1(0) \quad . \tag{5.46}$$

Again, we see that the variance and the weighting parameters control the exponential factor, and the bias is the main control in the multiplier factor, with the additional control, the ratio of the F_I and F_S weights.

We are now prepared to give some answer to the question of the neuromime's "memory". If a stimulus is presented and the g-sets altered according to that stimulus for a time t, then another stimulus is presented and the g-sets altered again, how much of the original adaptation remains?

The output of the neuromine at time t after the beginning of the adaptation is, from (5.1)

$$R(t) = M_I \vec{g}(t) \cdot \vec{x} + B \tag{5.47}$$

substituting (5.26) for $\vec{g}(t)$,

$$\begin{aligned} R(t) &= M_I \left[(e^{\lambda t} - 1) \frac{R(0)}{M_I \delta} \vec{u} + \vec{g}(0) \right] \cdot \vec{x} + B \\ &= (e^{\lambda t} - 1) \frac{R(0)}{\delta} \vec{u} \cdot \vec{x} + M_I \vec{g}(0) \cdot \vec{x} + B \\ &= R(0) e^{\lambda t} \end{aligned} \tag{5.48}$$

We will consider an input $\vec{x}_1$ followed by an input $\vec{x}_2$, and will then examine the effect of a repetition of $\vec{x}_1$. Parameters from (5.48) and (5.26) will be subscripted according to the input concerned. We have, from (5.48)

$$R_1(t) = R_1(0)e^{\lambda_1 t}$$

$$\vec{g}_1(t) = \frac{R_1(0)}{M_I \delta_1}(e^{\lambda_1 t} - 1)\vec{d}_1 + \vec{g}_1(0)$$

$$R_2(0) = M_I \vec{g}_1(t) \cdot \vec{x}_2 + B \qquad (5.49)$$

$$R_2(t) = R_2(0)e^{\lambda_2 t}$$

$$\vec{g}_2(t) = \frac{R_2(0)}{M_I \delta_2}(e^{\lambda_2 t} - 1)\vec{d}_2 + \vec{g}_1(t) \quad .$$

Applying the input $\vec{x}_1$ again, we have

$$R_3(0) = M_I \vec{g}_2(t) \cdot \vec{x}_1 + B \quad . \qquad (5.50)$$

The variable of main interest is the loss of adaptation

$$L = R_3(0) - R_1(t) \quad . \qquad (5.51)$$

Substituting the expressions in (5.50) and (5.49), into (5.51), we have

$$L = R_3(0) - R_1(t)$$

$$= M_I \vec{g}_2(t) \cdot \vec{x}_1 + B - R_1(t) \quad .$$

Substituting for $\vec{g}_2(t)$,

$$L = M_I\left[\frac{R_2(0)}{M_I\delta_2}(e^{\lambda_2 t} - 1)\vec{u}_2 + \vec{g}_1(t)\right] \cdot \vec{x}_1 + B - R_1(t)$$

$$= (M_I\vec{g}_1(t) \cdot \vec{x}_2 + B)(e^{\lambda_2 t} - 1)\frac{\vec{u}_2 \cdot \vec{x}_1}{\delta_2} + R_1(t) - R_1(t)$$

Substituting for $\vec{g}_1(t)$, noting that $R_1(t)$ drops out, and expanding,

$$L = \left\{\left[R_1(0)(e^{\lambda_1 t} - 1)\frac{\vec{u}_1}{\delta_1} + M_I\vec{g}_1(0)\right] \cdot x_2 + B\right\}(e^{\lambda_2 t} - 1)\frac{\vec{u}_2}{\delta_2} \cdot \vec{x}_1$$

$$= \left[R_1(0)(e^{\lambda_1 t} - 1)\frac{\vec{u}_1 \cdot \vec{x}_2}{\delta_1} + M_I\vec{g}_1(0) \cdot \vec{x}_2 + B\right](e^{\lambda_2 t} - 1)\frac{\vec{u}_2}{\delta_2} \cdot x_1 \cdot \tag{5.52}$$

From the formulation in Equation (4.26), we know that

$$\begin{aligned}\vec{u}_1 \cdot \vec{x}_2 &= \vec{u}_1 \cdot (\vec{u}_2 + \vec{m}_2) = \vec{u}_1 \cdot \vec{u}_2 \quad , \\ \vec{u}_2 \cdot \vec{x}_1 &= \vec{u}_2 \cdot (\vec{u}_1 + \vec{m}_1) = \vec{u}_2 \cdot \vec{u}_1 \quad .\end{aligned} \tag{5.53}$$

Defining

$$M_I\vec{g}_1(0) \cdot \vec{x}_2 + B = R'(0) \quad , \tag{5.54}$$

and

$$c_1 = (e^{\lambda_1 t} - 1) \quad , \quad c_2 = (e^{\lambda_2 t} - 1) \tag{5.55}$$

we have, substituting (5.53), (5.54) and (5.55) into (5.52),

$$L = \left[R_1(0)c_1\frac{\vec{u}_1 \cdot \vec{u}_2}{\delta_1} + R'(0)\right]c_2\frac{\vec{u}_2 \cdot \vec{u}_1}{\delta_2}$$

$$= R_1(0)c_1c_2\frac{(\vec{u}_1 \cdot \vec{u}_2)^2}{\delta_1\delta_2} + c_2R'(0)\frac{\vec{u}_1 \cdot \vec{u}_2}{\delta_2} \tag{5.56}$$

The dot product in (5.53) can be expressed in terms of the lengths of the vectors and the cosine of the angle between them [5] as

$$\vec{u}_1 \cdot \vec{u}_2 = |\vec{u}_1| \; |\vec{u}_2| \cos\theta \tag{5.57}$$

and since $\delta = |\vec{u}|^2$, $\vec{u}_1 \cdot \vec{u}_2 = \sqrt{\delta_1} \; \sqrt{\delta_2} \cos\theta$,

we have, from (5.56) if $R(0) = r_1$ and $R'(0) = r_2$,

$$L = r_1 c_1 c_2 \cos^2\theta + r_2 c_2 \sqrt{\frac{\delta_1}{\delta_2}} \cos\theta \tag{5.58}$$

The adaptation loss for $\vec{x}_1$ after having applied $\vec{x}_2$ depends, therefore, in great part on the angle between the two deviation vectors, which is reasonable since the adaptation of $\vec{g}$ is controlled by the deviation vector. Some insight into the distribution of L may be gained by looking at the distribution of $\cos\theta$ and $\cos^2\theta$. If we assume that the lengths of the deviation vectors are equal, we can imagine d_1 as the north pole of a 2n-dimensional hypersphere, and examine the angle θ between $\vec{d}_1$ and $\vec{d}_2$. An element of surface content of this hypersphere is derived in Appendix IV as

$$dS = (\rho d\theta)(\rho \sin\theta d\phi_1)(\rho \sin\theta \sin\phi_1 d\phi_2) \quad (\rho \sin\theta \sin\phi_1 \cdots \sin\phi_{2n-3} d\phi_{2n-2})$$

$$= \rho^{2n-1} \sin^{2n-2}\theta \sin^{2n-3}\phi_1 \cdots \sin\phi_{2n-3} d\phi d\phi_1 \cdots d\phi_{2n-2} \tag{5.59}$$

where ρ is the radius of the hypersphere, and $\theta, \phi_1, \ldots, \phi_{2n-3}$ are the successive hyperconical angles, corresponding to the latitude on a 3-dimensional sphere, ranging from 0 to π, while ϕ_{2n-2} is the planar angle, allowed to range from 0 to π .

The probability of θ occurring between 0 and θ_0 is the ratio of the surface content of a segment of the hypersphere cut out by

$$0<\theta<\theta_0 \quad , \quad 0<\phi_i<\pi \qquad\qquad i = 1,\ldots,2n-3, \; 0<\phi_{2n-2}<2\pi$$

which is proportional to, from (5.59), (with $\rho = 1$)

$$P(\theta<\theta_0) = \int_0^{\theta_0} \sin^{2n-2}\theta d\theta \int_0^{\pi} \sin^{2n-3}\phi_1 d\phi_1 \cdots \int_0^{\pi} \sin\phi_{2n-3} d\phi_{2n-3} \int_0^{2\pi} d\phi_{2n-2}$$

$$= K \int_0^{\theta_0} \sin^{2n-2}\theta d\theta \tag{5.60}$$

Since as n increases, $\sin^{2n-2}\theta$ decreases for $\theta \neq \pi/2$, the probability that θ deviates from $\pi/2$ decreases as n increases, and the averages of $\cos\theta$ and $\cos^2\theta$ approach 0. This lends justification to the belief that, on the average, the adaptation loss L will be small, so that a neuromime will tend to "remember" its previous training.

Section VI

STRUCTURE AND OPERATION OF NEUROMIME NETS

With the development in previous sections, we can now discuss in a unified manner the structure and behavior of a neuromime net, including some ideas on the use of the net simulators currently under development.

Recall that nets are organized as a series of computational areas or logical levels, each area containing a number of neuromime components. The S inputs of the components within an area come from the outputs of other components in the same area, while the I inputs come from outside the area, either from the primary sensory inputs or from the outputs of components in other computational areas.

A neuromime net computation area is composed of a set of computing elements

$$\{C_i \mid i=1,\ldots,m\}$$

together with their connections to the outside world. In addition, each component has a number of state inputs which are outputs from other components in the same area.

If r_j is the output from component C_j, the effect of the interaction is given by an iterative process, which proceeds until convergence is obtained. At time $i+1$, r_j is computed from the outputs of the other components at time i:

$$\begin{aligned} r_1(i+1) &= k_{12}r_2(i) + \cdots + k_{1m}r_m(i) + r_1(0) \\ r_2(i+1) &= k_{21}r_1(i) + k_{23}r_3(i) + \cdots + k_{2m}r_m(i) + r_2(0) \\ &\vdots \qquad\qquad\qquad \vdots \\ r_m(i+1) &= k_{m1}r_1(i) + \cdots + k_{m,\,m-1}R_{m-1}(i) + r_m(0) \end{aligned} \tag{6.1}$$

where the k_{ij} are coupling coefficients. The indices can be interpreted as the effect of the output of C_j on the output of C_i .

The system (6.1) is a linear system of equations, and can be re-expressed as

$$\rho(i + 1) = M\rho(i) + \rho(0) \tag{6.2}$$

where

$$\rho = \begin{pmatrix} r_1 \\ \vdots \\ r_m \end{pmatrix} \quad \text{and} \quad M = \begin{pmatrix} 0 & k_{12} & k_{13} & \cdots & k_{1m} \\ k_{21} & 0 & k_{23} & \cdots & k_{2m} \\ \vdots & \vdots & & & \vdots \\ k_{m1} & k_{m,\,m-1} & & & 0 \end{pmatrix}$$

and $\rho(0)$ is the set of outputs before the computation of the interaction has begun.
Since, at the beginning of the iteration,

$$\rho(1) = M\rho(0) + \rho(0) \tag{6.3}$$

(6.2) is by induction

$$\rho(i) = M^i\rho(0) + M^{i-1}\rho(0) + \cdots + M\rho(0) + \rho(0)$$

$$= \sum_{j=0}^{i} M^j \rho(0) \tag{6.4}$$

where M^0 is defined to be the identity matrix.

The example given in Figure 6.1 may help to clarify this process. The example illustrates the simple situation where $m = 2$. On the left are the symbolic interaction matrices for $i = 0, 1, 2, 3, 4$; and on the right are the corresponding matrices with $k_{12} = -.1$ and $k_{21} = .2$. At the bottom is given the sum of the five matrices.

M^i symbolic	i	M^i numeric
$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	0	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
$\begin{pmatrix} 0 & k_{12} \\ k_{21} & 0 \end{pmatrix}$	1	$\begin{pmatrix} 0 & -.1 \\ .2 & 0 \end{pmatrix}$
$\begin{pmatrix} k_{12}k_{21} & 0 \\ 0 & k_{21}k_{12} \end{pmatrix}$	2	$\begin{pmatrix} -.02 & 0 \\ 0 & -.02 \end{pmatrix}$
$\begin{pmatrix} 0 & k_{12}^2k_{21} \\ k_{21}^2k_{12} & 0 \end{pmatrix}$	3	$\begin{pmatrix} 0 & +.002 \\ -.004 & 0 \end{pmatrix}$
$\begin{pmatrix} (k_{12}k_{21})^2 & 0 \\ 0 & (k_{21}k_{12})^2 \end{pmatrix}$	4	$\begin{pmatrix} +.0004 & 0 \\ 0 & +.0004 \end{pmatrix}$
$\sum_{i=0}^{4} M^i$	=	$\begin{pmatrix} .9804 & -.098 \\ .196 & .9804 \end{pmatrix}$

Figure 3.1 Interaction Matrices of Dimension 2.

$$k_{12} = -.1 \ , \quad k_{21} = .2$$

If

$$\rho(4) = \sum_{i=0}^{4} M^i \rho(0) \tag{6.5}$$

then

$$r_1(4) = .9804\ r_1(0) - .098\ r_2(0) \tag{6.6}$$

$$r_2(4) = .196\ r_1(0) + .9804\ r_2(0) \tag{6.6}$$

As might be expected, the output from C_1 is decreased and the output from C_2 is usually increased, unless $r_1(0)$ is very small compared with $r_2(0)$, in which case the second order coupling of C_2 with itself, given by

$$(M^2)_{22} = k_1 k_2 = -.02 \tag{6.7}$$

becomes dominant.

A more complex example illustrating second order coupling more clearly, is given in Figure 6.2, showing M^2 for $m = 4$. Each entry in the matrix M^2, $(M^2)_{ij}$, is interpreted as the second order coupling of C_j on C_i. Specifically,

$$(M^2)_{11} = k_{12}k_{21} + k_{13}k_{31} + k_{14}k_{41} \tag{6.8}$$

is the coupling of C_1 on itself through C_2, plus the coupling of C_1 on itself through C_3, plus the coupling of C_1 on itself through C_4. Similarly,

$$(M^2)_{34} = k_{31}k_{14} + k_{32}k_{24} \tag{6.9}$$

is the coupling of C_4 on C_3 through C_1 plus the coupling of C_4 on C_3 through C_2.

A problem of importance is whether the system will eventually produce a stable output, which is the same as determining whether

$$\lim_{i\to\infty} \sum_{j=0}^{i} M^j \equiv M^*$$

$$M = \begin{pmatrix} 0 & k_{12} & k_{13} & k_{14} \\ k_{21} & 0 & k_{23} & k_{24} \\ k_{31} & k_{32} & 0 & k_{34} \\ k_{41} & k_{42} & k_{43} & 0 \end{pmatrix}$$

$$M^2 = \begin{pmatrix} [k_{12}k_{21}+k_{13}k_{31}+k_{14}k_{41}] & [k_{13}k_{32}+k_{14}k_{42}] & [k_{12}k_{23}+k_{14}k_{43}] & [k_{12}k_{24}+k_{13}k_{34}] \\ [k_{21}k_{13}+k_{23}k_{31}] & [k_{21}k_{12}+k_{23}k_{32}+k_{24}k_{42}] & [k_{21}k_{13}+k_{24}k_{43}] & [k_{21}k_{14}+k_{23}k_{34}] \\ [k_{32}k_{21}+k_{34}k_{41}] & [k_{31}k_{12}+k_{34}k_{42}] & [k_{31}k_{13}+k_{32}k_{23}+k_{34}k_{43}] & [k_{31}k_{14}+k_{32}k_{24}] \\ [k_{42}k_{21}+k_{43}k_{31}] & [k_{41}k_{12}+k_{43}k_{32}] & [k_{41}k_{13}+k_{42}k_{23}] & [k_{41}k_{14}+k_{42}k_{24}+k_{43}k_{34}] \end{pmatrix}$$

Figure 6.2
Interaction Matrices of Dimension
4 for $i = 1$ and $i = 2$

exists. It has been demonstrated in Appendix V that the system will be stable if the determinant

$$|I - M| > 0 \tag{6.11}$$

One scheme for interconnecting elements so as to obtain the most flexible coupling coefficients is that discussed in Figure 2.5, where r_j is connected to both the ℓ^{th} S^+ and S^- inputs of C_i. The interaction given by

$$r_j g_\ell^+ - r_j g_\ell^- = r_j(g_\ell^+ - g_\ell^-) \tag{6.12}$$

and the coupling coefficient is

$$k_{ij} = M_S(g_\ell^+ - g_\ell^-) \tag{6.13}$$

It is now possible to state in a unified way the operations performed by a net of m components each of which has n inputs. We require a number of definitions, and will re-define some of the symbols used in previous sections to obtain a clearer statement.

Let N be the incidence matrix of the net, which describes its interconnection structure.

$$N = (n_{ij})$$

where $n_{ij} = 1$ if the output of C_j is connected to the state input of C_i. N is an $m \times m$ matrix.

The inputs to and parameters of each component will be included in a series of $m \times n$ matrices, the i^{th} row of which will correspond to the associated vector of the i^{th} component. These matrices are defined in Table 6.1.

Table 6.1

Variable Matrices for Net Level

m×n Matrix	i^{th} row
X	the inputs to C_i ordered as in (4.19)
$\bar{X}$	the vector $\vec{m}$ of C_i as in (4.19)
G	the g-set of C_i as in (4.19)
D	the vector $\vec{d}$ of C_i as in (4.19)

Parameters common to the net are:

β - vector each of whose m components is the bias parameter B of (2.1)

w_1 - the parameter M_I of (2.1)

w_2 - the parameter M_S of (2.1)

There must be a set of deviation vectors for the interaction inputs at each component. These can be obtained from the output vector ρ and the incidence matrix N as follows:

$$E_{ij} = N_{ij}\left[r_j - \frac{\sum_{\ell=1}^{m} N_{i\ell} r_\ell}{\sum_{\ell=1}^{m} N_{i\ell}}\right] \qquad (6.14)$$

The incidence matrix acts to pick out those outputs r_ℓ which are attached to component C_i, and the sum across row i of N is the number of r_ℓ attached to C_i.

The result of adaptation may be expressed by multiplying each row of the deviation matrices by an "adaptation coefficient" determined as in Section 5. These coefficients will be gathered in the vectors λ_1 and λ_2, coefficients for the external inputs and state inputs, respectively. The operation of multiplying the i^{th} row of a matrix A by the i^{th} component of a vector α where A has p rows and α has p elements, will be denoted by

$$B = \alpha \circledast A \tag{6.15}$$

Another operation used is

$$\alpha = \text{diag}(A) \tag{6.16}$$

α is the vector whose elements are the diagonal elements of the matrix A.

We can now state the sequence of operations of a net. First, the outputs are computed:

$$T = w_1 X$$

$$M^* = \sum_{i=0}^{\infty} M^i \tag{6.17}$$

$$\rho = M^*[\text{diag}(TG^t) + \beta]$$

Next, the matrices M and G are altered according to the adaptation:

$$G' = \lambda_1 \circledast D + G$$

$$M' = w_2\lambda_2 \circledast E + M$$

Experimental nets can be constructed and tested using the simulators under development. The general purpose digital simulator is currently operational, and its structure is closely related to the theoretical structure developed in this report.

An important property of this simulation is that nets can be constructed on a statistical basis, freeing the programmer from tediously specifying the interconnections among all of the components on an individual basis. In this construction, a metrical structure is imposed on the net so that a distance is defined from one component to another component on the same or a different level, including a distance from the primary sensory inputs on the first level. A standard distance function is available, but the programmer may arbitrarily define a new one if he wishes.

The program generates a net, making connections between components on a probabilistic basis, the probability depending on the distance in some fashion. There are several standard probability functions in the program, and the programmer may also define new ones.

Another feature of this simulator is that it has a criterion for the activity level of a computational area, and automatically adjusts the bias level to keep the activity within a certain pre-established range. This device controls the relative importance of the communication channels between different areas, so that certain information paths are given more weight than others on an a priori basis. It also functions as an overall gain control.

Section VII

UNSOLVED PROBLEMS

During the course of this program, a number of interesting problems were formulated but not solved. These problems motivated the development of the formalism of this report, and should be amenable to the attack of these new tools. In addition, a number of extensions to the formalism were suggested as the work proceeded, but were not on the main line of endeavor, so are as yet undone. This section is a compendium of these ideas, and they will form a basis for further research.

Our knowledge of the theoretical properties of neuromime nets must be put to an experimental test to ensure that the model is valid. A natural test is to make use of data generated by the laboratory's elaborate sound generating and anditory system simulators.

Since the currently available neuromime net simulation equipment does not allow time varying inputs, we must restrict our attention to quantities which are either independent of starting time or can be synchronized in the sampling apparatus. This restriction limits us to considering only fixed frequency transient signals or very noisy or periodical high frequency signals, since the former can be generated with associated synchronization pulse, and the latter can be considered as stationary time series and has observable properties independent of the starting time.

A reasonable series of experiments would therefore begin by attempting to differentiate simple transients, progressing to more complex transients. An attempt could then be made to make equivalent certain sets of complex transients while differentiating between sets. Later experimentation would include high frequency continuous signals.

An important aspect of this series of experiments is an investigation of the coupling between the neuromime net and its environment, especially from the learning point of view. We need to know how the sequences of stimuli should be presented and how we should reinforce the network. The work of Rosenblatt (1962) on the perceptron will be of great assistance in this area. It is likely that neuromime nets have information processing capabilities equal to or better than perceptrons, so we should be able to apply many of the learning theorems which he has derived.

Additional work should be done on this model in the areas of statistical properties of interconnection matrices, both internal and external. It would seem that concepts of information theory could be employed to good effect, especially if related to the topological structure of the net.

A formulation which might be of use in this problem is that of channel capacity density. We can define a cross section of an information channel in such a way as to identify sets of points which a signal propagating through the channel will occupy simultaneously. Some examples are given in Figure 7.1. Part (a) shows a transmission line, where the cross section is a point. Part (b) shows transmission along a plane, the cross section is a line. Part (c) shows the transmission through a network, the cross sections are sets of points.

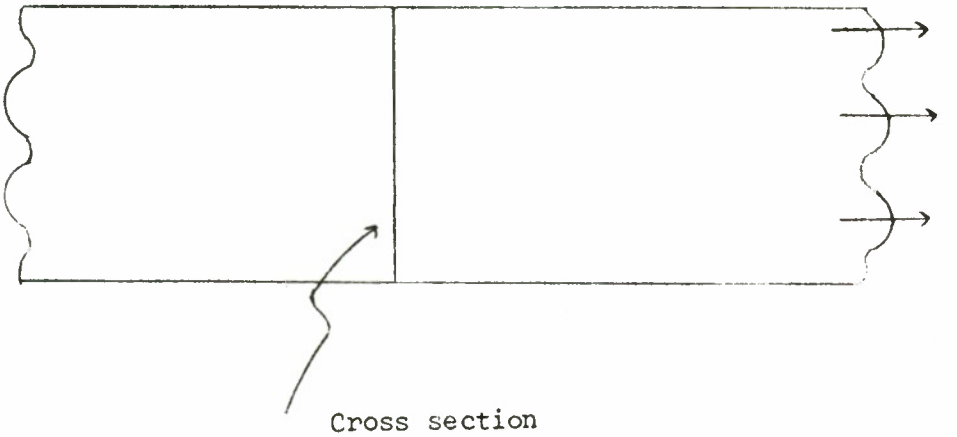
If we defined a function giving channel capacity per unit of cross section, we can consider this a channel capacity density function. By examining this function, we should be able to derive many statistical properties of neuromime net information handling. The work of Inselberg, Von Foerster, et al (1963) is relevant to this area.

The control theoretic formulation of the interactions in a net should produce some more useful information. Appendix V discusses this area in more detail. A possible approach here would be to combine the statistical interconnections with the control formulation, and develop a statistical control theory.

Implicit in the discussion in Section III is that a neuromime net can be considered as a sort of flexible measuring device on classes of input stimuli, measuring the presence of certain patterns, and recalibrating itself as it proceeds. The output of a net can be considered as a sort of activity layer, where activity is localized in spots corresponding to particular patterns, as in Figure 7.2(a). If the net can be trained to respond to a new pattern, while keeping its response to the old ones, a new output area might be formed, displacing some of the others, as in Figure 7.2(b). It would seem that this property of Neuromime net processing could be used to develop a theory of the operations performed on patterned sets of stimuli.

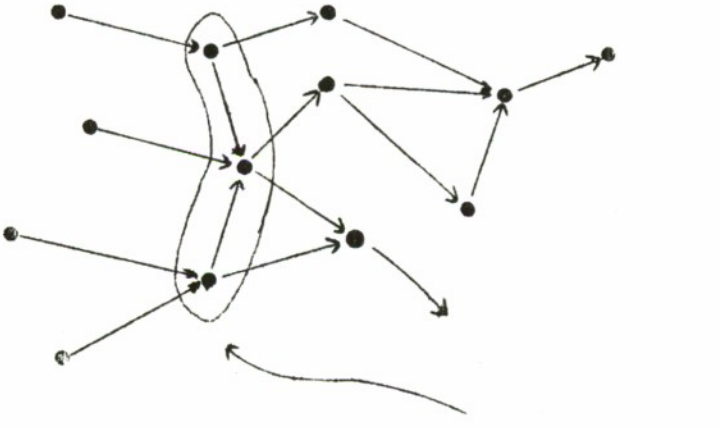
(a)

Line transmission



(b)

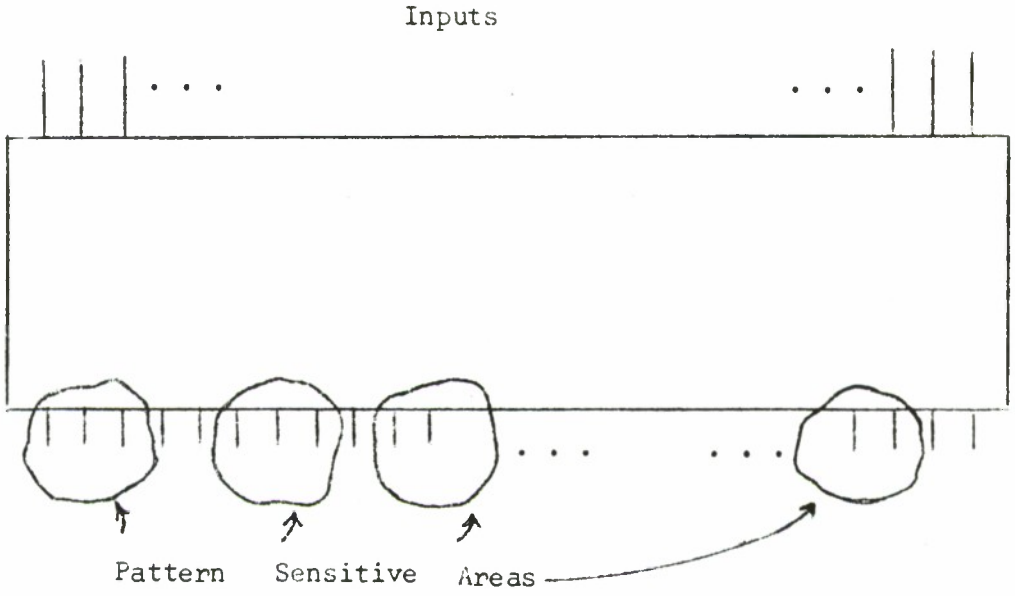
Homogeneous Plane Transmission



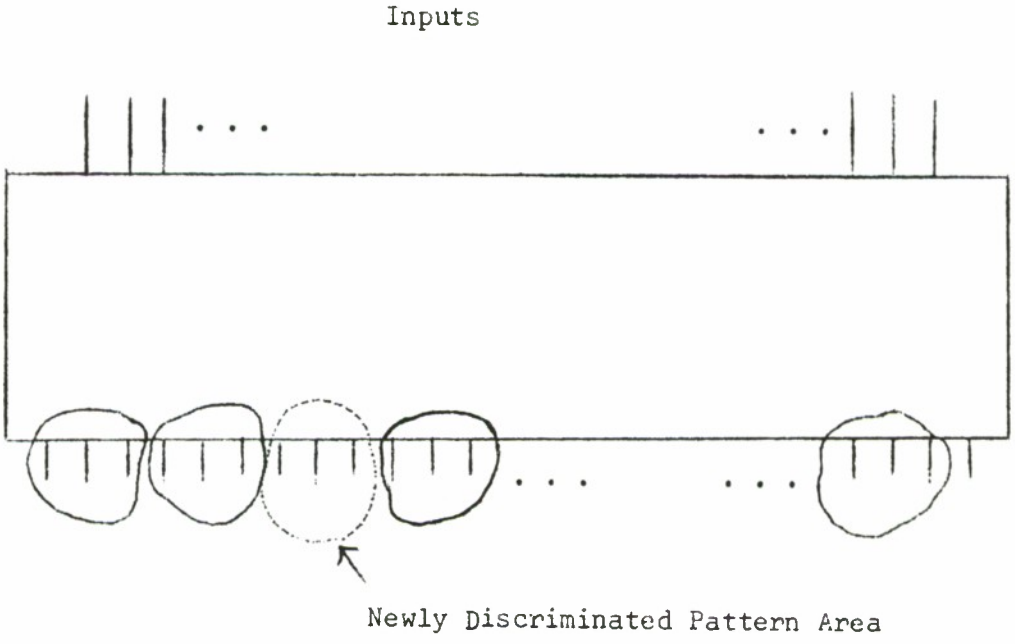
(c)

Network Transmission

Figure 7.1. Channel Cross-sections



a. Original Pattern Sensitive Areas



b. Addition of New Area

Figure 7.2. Neuromime Nets as Variable Measuring Devices

Properties of patterns that are relevant to the above structure are that they require both a knowledge of the outputs from a set of sensors and a knowledge of the location of these sensors in the outside world. One may consider a sensory field with a topological structure as a window on the world. The pattern measurement device has a number of different kinds of sensors, and these are attached to the field, sometimes with several different kinds attached to the same area. A pattern is defined with two characteristics, a particular configuration of outputs from a set of sensors, and a particular topological arrangement of that set of sensors.

In most cases, it is useful to have the pattern meter measure the degree of presence or the probability of presence of a pattern rather than make the outright statement that the pattern is or is not present. For this reason, the pattern meter must measure not only how near is the sensory output configuration to a standard, but also how near is the topological arrangement of the sensors to its standard. For example, a straight line in a visual field is determined both by a constant contrast level in a set of sensors, and by the nearness to a straight line of the sensors exhibiting the desired configuration of outputs. An observer would say that the stimulus in Figure 7.3(a) was a straight line, and would say also that the stimuli in (b), (c), and (d) were nearly straight lines.

A neuromime component can, in itself, adjust the particular subfield of the sensory field it observes (by setting some of its g-weights to zero), and its output gives a measure of nearness of the input stimulus to a standard, so we can hope that neuromime nets can be constructed to measure both kinds of distances.

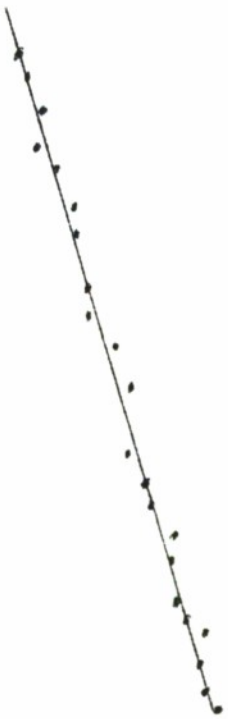
Besides the foregoing investigation of the properties of neuromimes and nets which can be simulated by currently available programs, there are extensions that should be made in the theoretical and conceptual structure to give the study more direction.

One such extension is the reformulation of the neuromime equations so that they make sense with rapidly time-varying inputs. It would be useful if the time-varying formulation reduced to the constant formulation when the inputs were constant.

a. Straight Line

b. Some Points Missing
(deficient in attributes)

c. Points off Line
(Deficient in Geometry)



d. Both deficiencies

Figure 7.3. Nearly Straight Line Stimuli

A possible approach to this problem is linear prediction as described by Wiener (1949) and extended by Allais (1964). Assume we have a random function $f(t)$, and we wish to predict the value of $f(t_0 + \tau)$ for some future increment τ, where t_0 is the current time. We are free to use any information from $-\infty$ to t_0 to determine $f(t_0 + \tau)$. Wiener, therefore, defines a Kernel function $K(t,\tau)$, and a predictor

$$P(t, \tau) = \int_{-\infty}^{t_0} K(t, \tau)\, f(t)dt \tag{7.1}$$

The goodness of the predictor is given by the mean square error

$$\varepsilon(t_0, \tau) = E\{[P(t_0, \tau) - f(t_0 + \tau)]^2\} \tag{7.2}$$

According to Allais, there is no reason why we must confine ourselves to predicting the future values of $f(t)$. This prediction theory applies equally well to any function $h(f, t)$ which is correlated with f; for example, f could be an electrocardiogram, and h could presumably be a function which is zero before the QRS complex and one after it, or, more interesting, a function which is one when the QRS complex passes a certain point in time, and zero otherwise.

It would thus seem that if the g-sets were defined as predictor functions, so that $g(t)$ is a kernel function defined on the interval $(-\infty, t_0)$, and the output is computed by

$$R(t_0) = \int_{-\infty}^{t_0} \vec{g}(t) \cdot \vec{x}(t)dt \tag{7.3}$$

$$= \sum_{i=1}^{2n} \int_{-\infty}^{t_0} g_i(t) \cdot x_i(t)dt$$

we could treat the time-varying case. The problem of adapting the g-weights could be handled in some such way as setting

$$\vec{g}(t) = (a\vec{g}(t) + (1 - a)\,\vec{x}(t))R(t_0) \qquad (7.4)$$

for $(0 < a < 1)$, but this is a problem which merits some thought.

Note that the computation of R is similar to the constant case as in (2.2). Component interaction could perhaps be defined in the same way as in (2.2) since the feedback characteristics of a net are independent of the input functions (See Appendix V).

A beginning on a theory of organization was made during this work. It employs a value system--a hierarchy of goals which the system is to pursue. A sample system is given in Figure 7.4. Some of the goals contribute to others, and some are independent of each other.

A single goal-seeking unit is a mechanism which monitors a series of inputs to determine the state of the world, performs some computations, and issues a set of instructions to the set of mechanisms available to it to change the state of the world. (The world can include the system itself.) A schematic diagram of a box appears in Figure 7.5.

The key concept here is that the world can be thought of as a space of states (perhaps a vector space). It is well known that a representation of a vector in the form $(a_1, a_2, \ldots, a_n)$ means $a_1\alpha_1 + a_2\alpha_2 + a_3\alpha_3 + \cdots + a_n\alpha_n$, where $(\alpha_1,\ldots,\alpha_n)$ are a set of reference vectors, usually an orthonormal basis. We can consider the sensory input vector $\vec{I}$ to be a representation of the world in the set of reference vectors $(\alpha_I^1, \alpha_I^2,\ldots,\alpha_I^n)$, where each α_I^k can be considered to refer to that portion of the world "seen" by the kth sensory input mechanism. The reference vectors $\{\alpha_I^k\}$ are not necessarily linearly independent. Similarly, the set $\vec{C}$ of commands to the system's effector devices is a vector, represented in a set of reference vectors $(\alpha_C^1,\ldots,\alpha_C^m)$, where each α_C^k represents the change in the world produced by a unit instruction to a particular effector device.

The goal in the goal-seeking unit is represented by a vector $\vec{G}$ which is the desired state of the external world as represented by the input mechanism. The instruction vector is computed from the deviation between the actual inputs and the desired inputs.

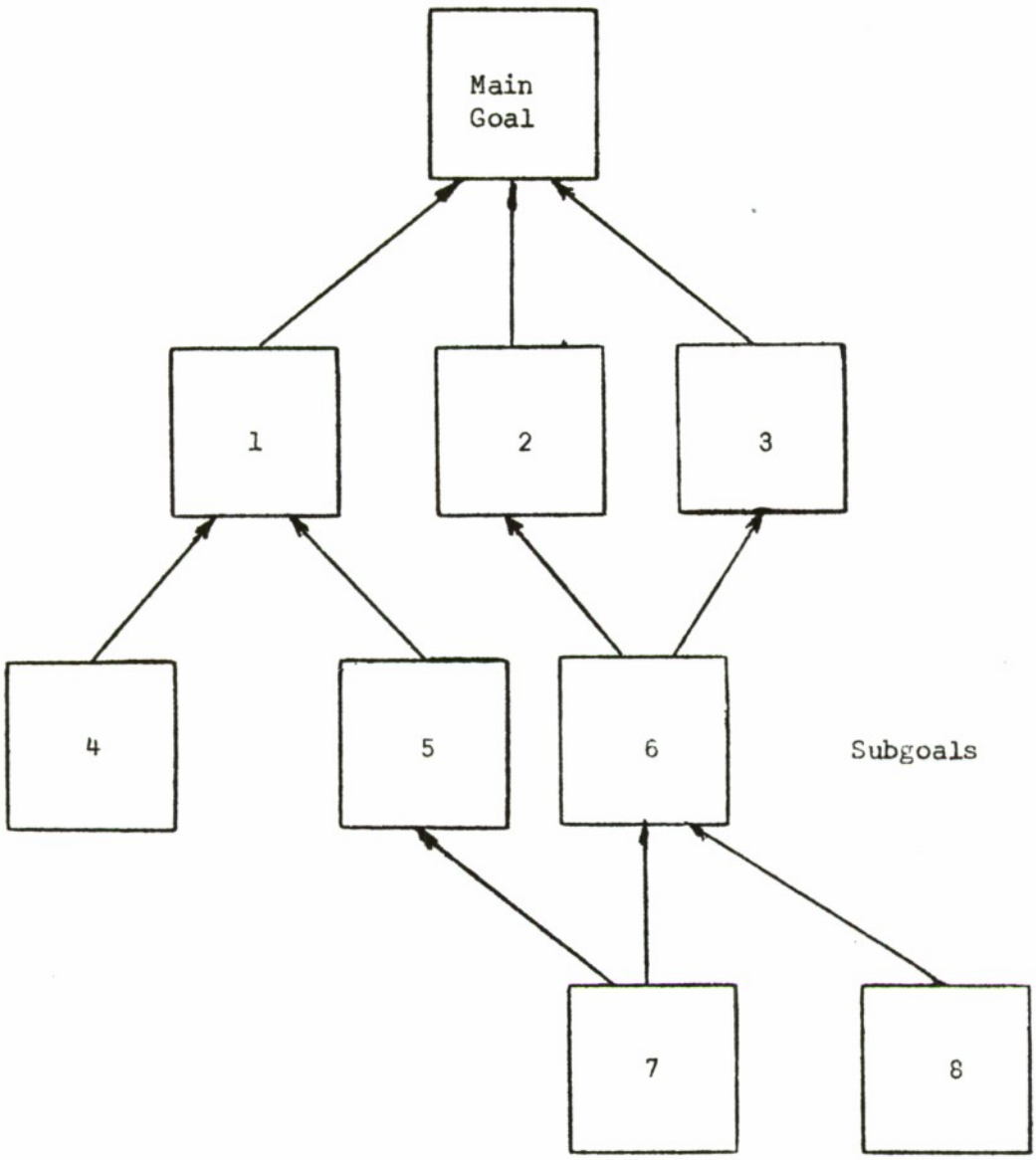


Figure 7.4. Sample Goal Lattice

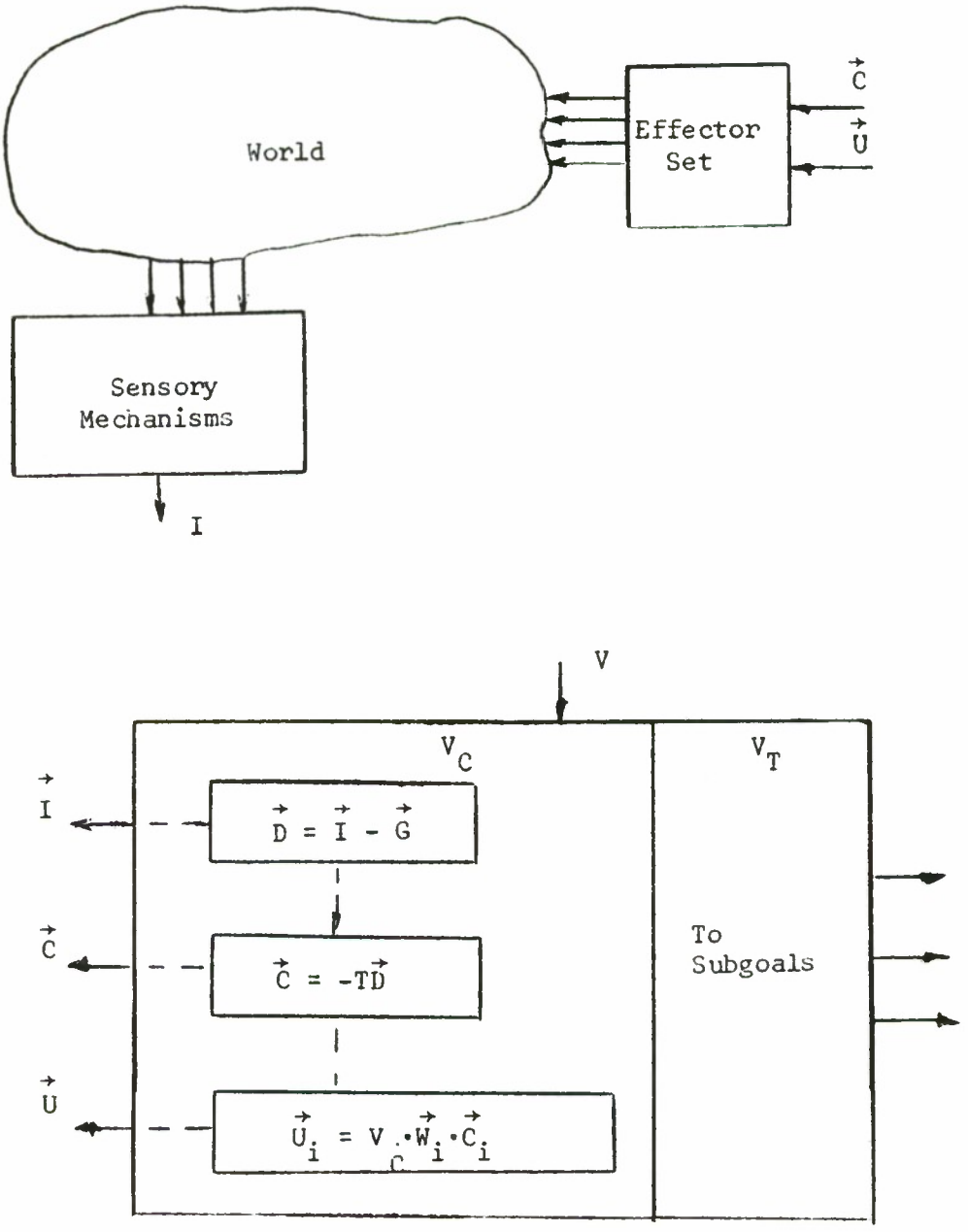


Figure 7.5. Goal-seeking Unit with its Connection to the World

$$\vec{D} = \vec{I} - \vec{G}. \tag{7.5}$$

The desired action is to alter the world so that $\vec{D}$ will become $\vec{0}$, so that the set of instructions to the effector set will represent a change of state of the external world equal to the deviation of the input vector $\vec{I}$ from the goal. This can be done by considering an $m \times n$ transform T mapping $\{\alpha_I\}$ into $\{\alpha_C\}$, so that

$$\vec{C} = -T\vec{D}. \tag{7.6}$$

T is not necessarily a linear transform.

Competition between various goals is introduced when contradictory instructions are given to the effector set. To deal with this, it is necessary to introduce a system of priorities. This is done by assigning a unit value V to the primary goal. This value is distributed by the primary goal to its subgoals, which in turn distribute value further down the lattice. Sometimes a goal will receive value from two or more supergoals, in which case the values can be considered to add together.

Not all of the value is transmitted, however. If a goal-seeking unit has a capacity to give instructions to the effector set (some may not--acting as logical goals and leaving all action to subgoals), some of its value must be assigned to its commands. This is represented by V_C in Figure 7.5. (Transmitted value is represented by V_T.)

A possible distribution system is to construct a vector $\vec{U}$ where

$$\vec{U}_i = V_C \vec{C}_i \vec{W}_i. \tag{7.7}$$

$\vec{W}$ is a weighting vector which normalizes the utility of each individual instruction. $\vec{W}$ is assigned so that

$$\sum_{i=1}^{n} U_i \leq V_C \tag{7.8}$$

for all $\vec{C}$ which can occur in practice. This is to allow for assignment of large value to emergency goals which are allowed to override normal goals when need arises. Usually the $\vec{U}$ vector will be small enough so that other instructions will take precedence.

Precedence is decided by the effector mechanism concerned by some rule according to the nature of its action. If it can execute only one instruction at a time, it will execute the one with greatest value. If it is a distribution mechanism it can fill orders in order of value, in proportion to the product of need and value, or in some other manner.

This sort of organization structure would be the organization of a "bug", an organism completely pre-programmed to behave in a complicated way. Neuromime nets are adaptive mechanisms, so it is reasonable to expect that devices could be made with some of the following abilities.

1. For a goal-seeking unit to construct its own T.

2. For the system to distribute value throughout itself so as to optimize its behavior in some way.

3. For a unit to seek new inputs if it cannot differentiate between signals that it must differentiate, or throw away inputs if it does not need them.

4. If two units find themselves in competition to mutual disadvantage, for them to form a coalition by generating a new super box which can regulate them in some way.

5. For the system to be able to modify its goal-seeking units, even to the extent of adding or deleting entire units.

Possible mechanisms for such adaptation might be gained by looking at some natural mechanisms. One such mechanism, homeostasis, has been considered in a primitive way by Ashby [11]. This is the tendency for a system to attempt to return to an equilibrium state after being perturbed. It is the mechanism behind the computations of a goal-seeking unit as described previously.

Another mechanism which might be applicable is that possibly used by nature in evolution. The parameters of the system are changed at random and the resulting system evaluated for goodness of performance in some way. If it is better than the old, it replaces the old, being otherwise discarded.

A conceptual structure which might cast some light on the organization of such a system without a homunculus is Wiener's []2] concept of the brain as a set of coupled nonlinear oscillators. This mathematical formalism might also be relevant to the design of neuromimes with time-varying inputs.

APPENDIX I

SOME PROPERTIES
OF THE n-DIMENSIONAL OCTAHEDRON

The development will be largely intuitive and what arguments are given will be largely combinatorial in nature. For those unfamiliar with n-dimensional geometry, a good mathematical treatment is given in An Introduction to the Geometry of N-Dimensions by D. M. Y. Sommerfield, Dover, 1958.

The Fourth Dimension Simply Explained edited by Henry P. Manning, Dover, 1960, presents several intuitive ways of imagining higher dimensional spaces.

The n-dimensional octahedron may be defined to be that figure bounding the region of space determined by n mutually perpendicular coordinate axes $x_1, \ldots, x_n$ fulfilling the condition

$$|x_1| + \cdots + |x_n| \leq 1.$$

The boundary is an (n-1)-dimensional surface determined by the equation

$$|x_1| + \cdots + |x_n| = 1.$$

This concept may be developed intuitively by considering the cases where $n = 2$ and $n = 3$. In the former, the boundary is the 1-dimensional surface

$$|x_1| + |x_2| = 1,$$

in the latter the boundary is the 2-dimensional surface

$$|x_1| + |x_2| + |x_3| = 1$$

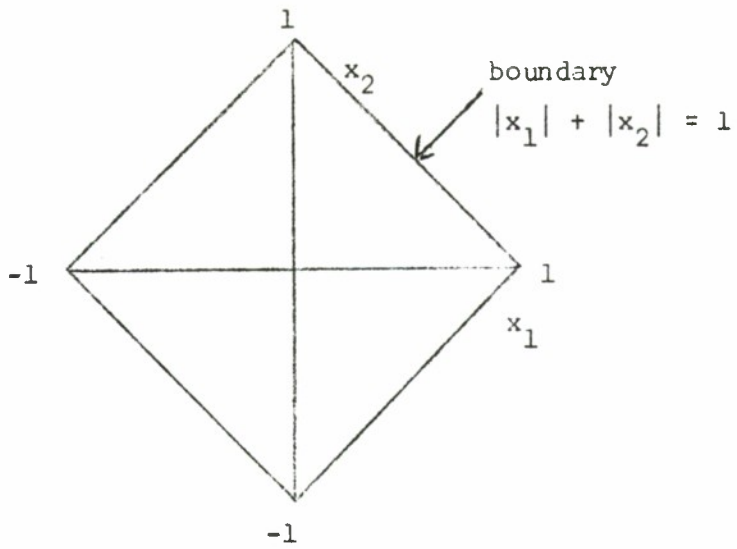
(see Figure 1).

If we restrict the range of the x_i so that $0 \leq x_i \leq 1$, we obtain the <u>principal section</u>, whose boundary is given by the coordinate planes and the (n-1)-dimensional hyperplane determined by the boundary equation, which becomes

$$x_1 + x_2 + \cdots + x_n = 1.$$

This hyperplane is defined as an <u>nth order face</u> of the octahedron. There are clearly 2^n nth order faces, since it is possible to divide the interior into 2^n sections by restricting some of the x_i to the range $-1 \leq x_i \leq 0$ and the others to the range $0 \leq x_i \leq 1$. (Each x_i can be restricted to either of two ranges, and there are n of the x_i's.)

Two-dimensional Octahedron



Three-dimensional Octahedron

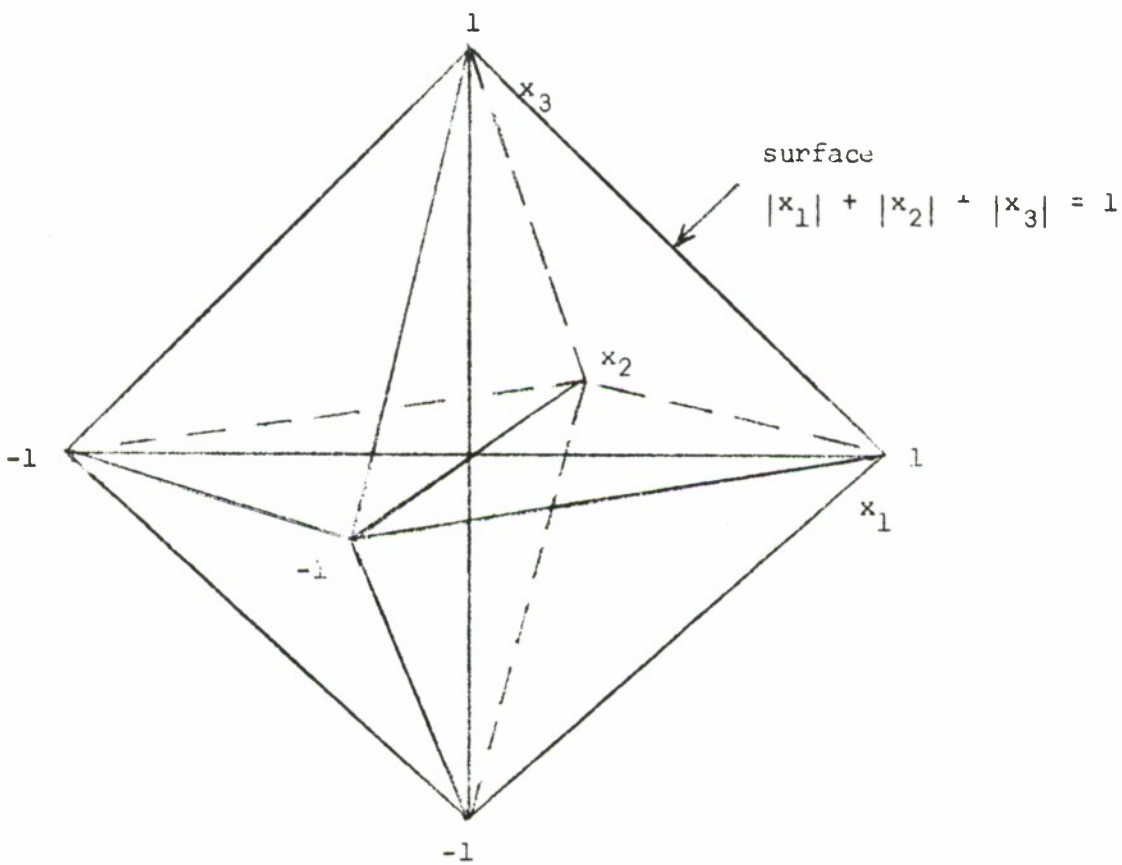


Figure 1

In each section a set of vertices, $\{v_1, \ldots, v_n\}$, is defined by the intersection of the face with the coordinate axes. These vertices define a region in the (n-1)-dimensional hyperplane, and this region is an (n-1)-dimensional simplex with its interior. This corresponds to a triangle in 2-dimensional space and a tetrahedron in 3-space. The edges of the simplex are the set of lines connecting the vertices taken two at a time. These edges are all of equal length, since each one forms the hypotenuse of a right triangle whose sides equal 1, so that the simplex is a regular simplex. This corresponds to an equilateral triangle in 2-space and a regular tetrahedron in 3-space. (See Figure 2.)

The regular simplex has the property that if any r of its vertices are taken away, the remaining $n - r$ vertices define an (n-r-1)-dimensional simplex. This corresponds to taking the intersection of the nth order face with the (n-r-1)-dimensional hyperplane containing $n - r$ of the n coordinate axes. For $n = 4$ and $r = 1$ this corresponds to the intersection of the hyperplane containing, say, the x_1, x_2, and x_3 coordinate axes with the tetrahedron (v_1, v_2, v_3, v_4) of Figure 2, resulting in the equilateral triangle (v_1, v_2, v_3).

We will call the intersection with the nth order face of a (k-1)-dimensional hyperplane containing k vertices in the principal section a kth order face. This is a (k-1)-dimensional simplex containing k of the n vertices.

Since a kth order face is a slice, as it were, of the nth order face, the points of both its boundary and interior satisfy the boundary equation (see Figure 2).

It is easy to count the number of kth order faces in an n-dimensional octahedron. A single kth order face is a simplex whose vertices are k of the n vertices of an nth order face. The number of combinations of n things taken k at a time is

$$\binom{n}{k} = \frac{n!}{k!(n - k)!} .$$

There are 2^n nth order faces, and each k^{th} order face is in common to 2^{n-k} sections, making $\binom{n}{k}2^k$

kth order faces in the entire octahedron. (A first-order face is defined to be a vertex, and a second-order face is an edge.)

It should be noted that a kth order face is a face of a k-dimensional octahedron as well.

We are going to want to divide the principal section into regions. To do this, it is useful to consider the problem in the hyperplane of the face. We can orient ourselves in this hyperplane by considering the location of various solutions to the boundary equation.

Consider the case with $n = 4$. The boundary equation is

$$x_1 + x_2 + x_3 + x_4 = 1$$

and this defines the interior and boundary of a regular tetrahedron, as in Figures 2 and 3. Figure 3 is a map of this tetrahedron. The points are given as quadruplets, having the values of the coordinates x_1, x_2, x_3, x_4 in order.

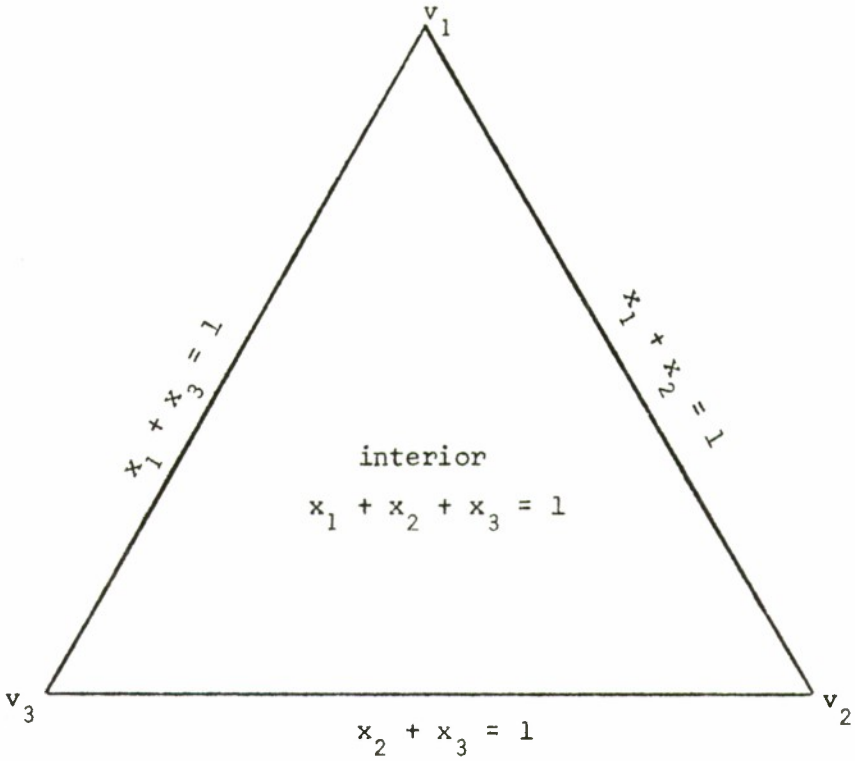
The partitioning to be considered in detail is that defined by the inequalities $x_i > x_j$. In other words, we want to take every point whose x_i coordinate is greater than its x_j coordinate, the other coordinates unconstrained, put them into a region, and put those points not fulfilling that condition into another. The boundary between the two regions is the hyperplane $x_i = x_j$. This is shown in the 2-dimensional and 3-dimensional case in Figure 4, and in the 4-dimensional case in Figure 5.

We note that the set of hyperplanes $x_i = x_j$ separate every kth order face contained in the nth order face bounding the principal section. This is true since two kth order faces are determined by the equations:

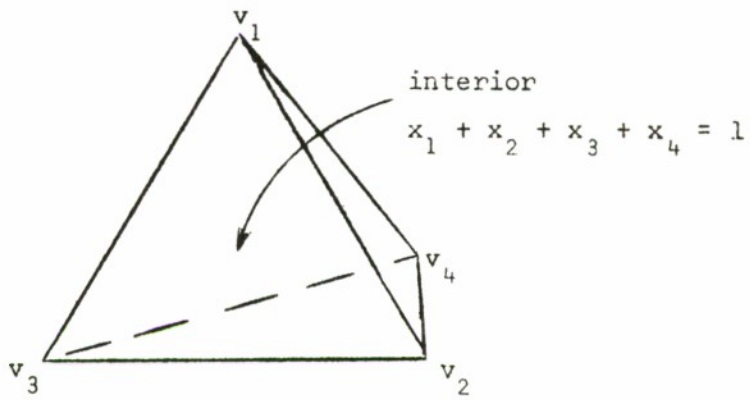
$$x_{i_1} + \cdots + x_{i_k} = 1 \qquad x_i = 0 \quad \text{for} \quad i \notin \{i_1, \ldots, i_k\}$$

$$x_{j_1} + \cdots + x_{j_k} = 1 \qquad x_j = 0 \quad \text{for} \quad j \notin \{j_1, \ldots, j_k\}$$

where there is at least one pair (x_i, x_j) not in common. The edge (x_i, x_j) of the nth order face is bisected by the hyperplane $x_i = x_j$, separating the two faces.

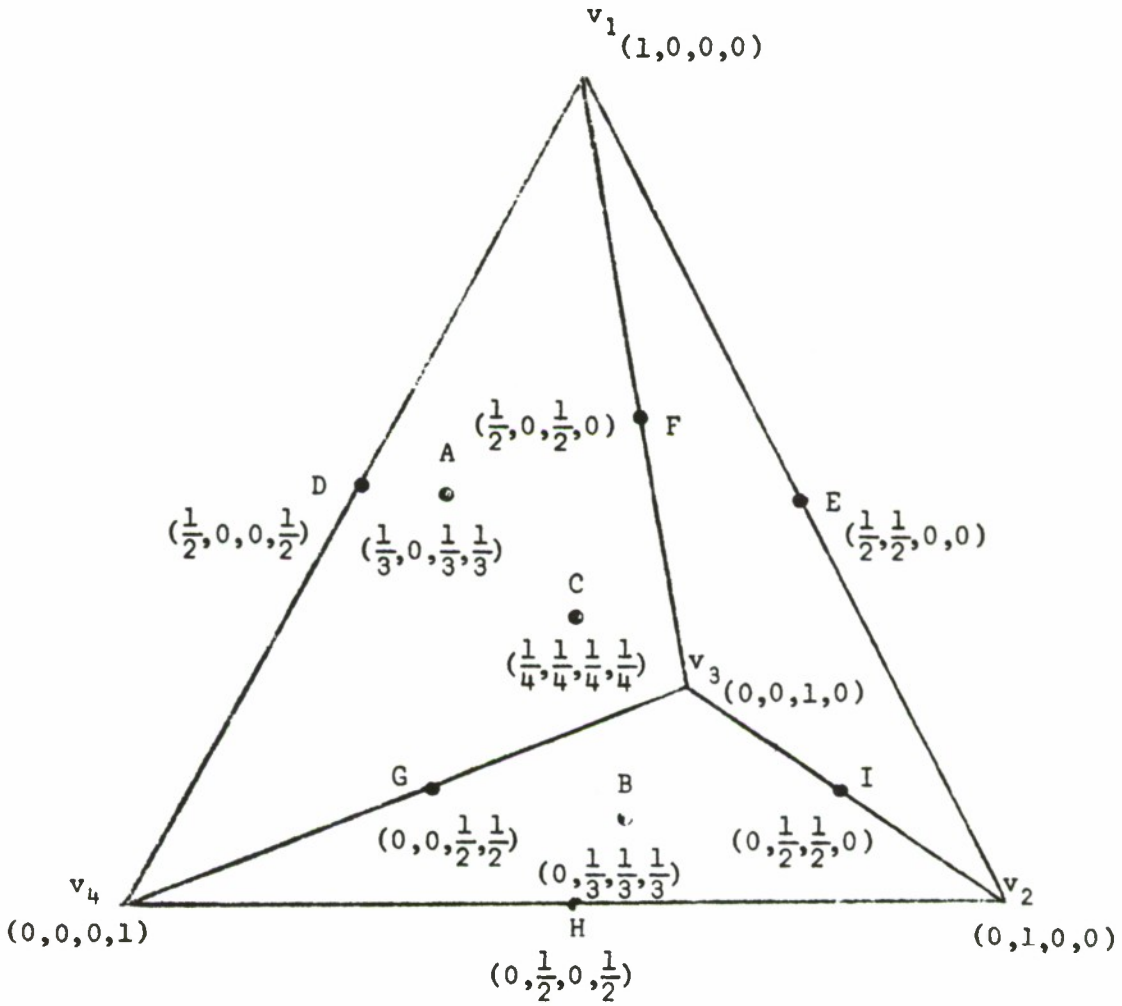


Two-dimensional Simplex (Equilateral Triangle)



Three-dimensional Simplex (Regular Tetrahedron)

Figure 2



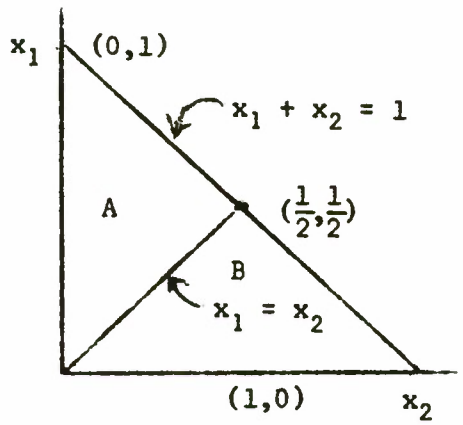
A - center of triangle $v_1v_3v_4$

B - center of triangle $v_2v_3v_4$

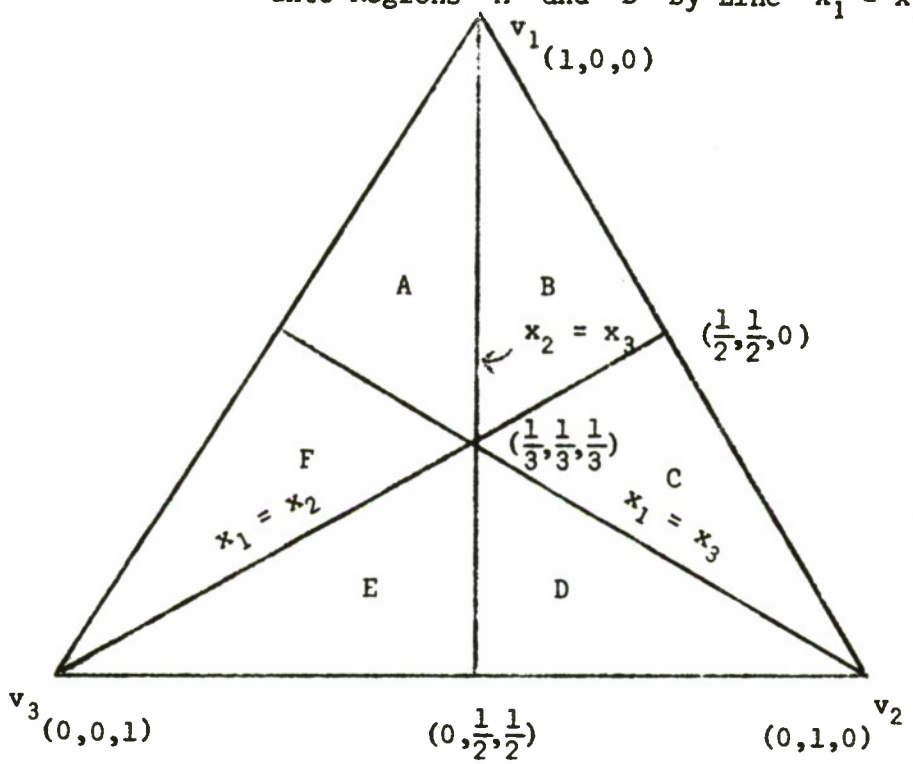
C - center of tetrahedron

D - I - center of edges as indicated

Figure 3. Map of Tetrahedron



Two-dimensional Case--Plane Divided Into Regions A and B by Line $x_1 = x_2$



Three-dimensional Case--Showing Six Regions in Plane of Face, Above Below, Stage in Construction

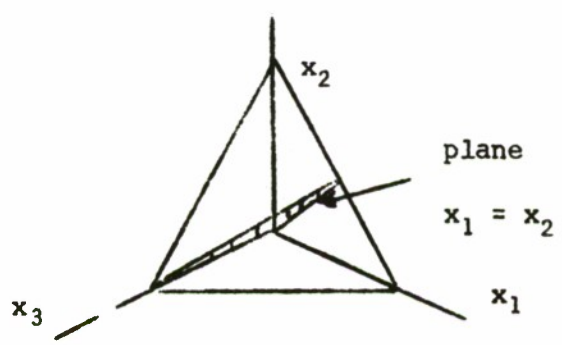


Figure 4

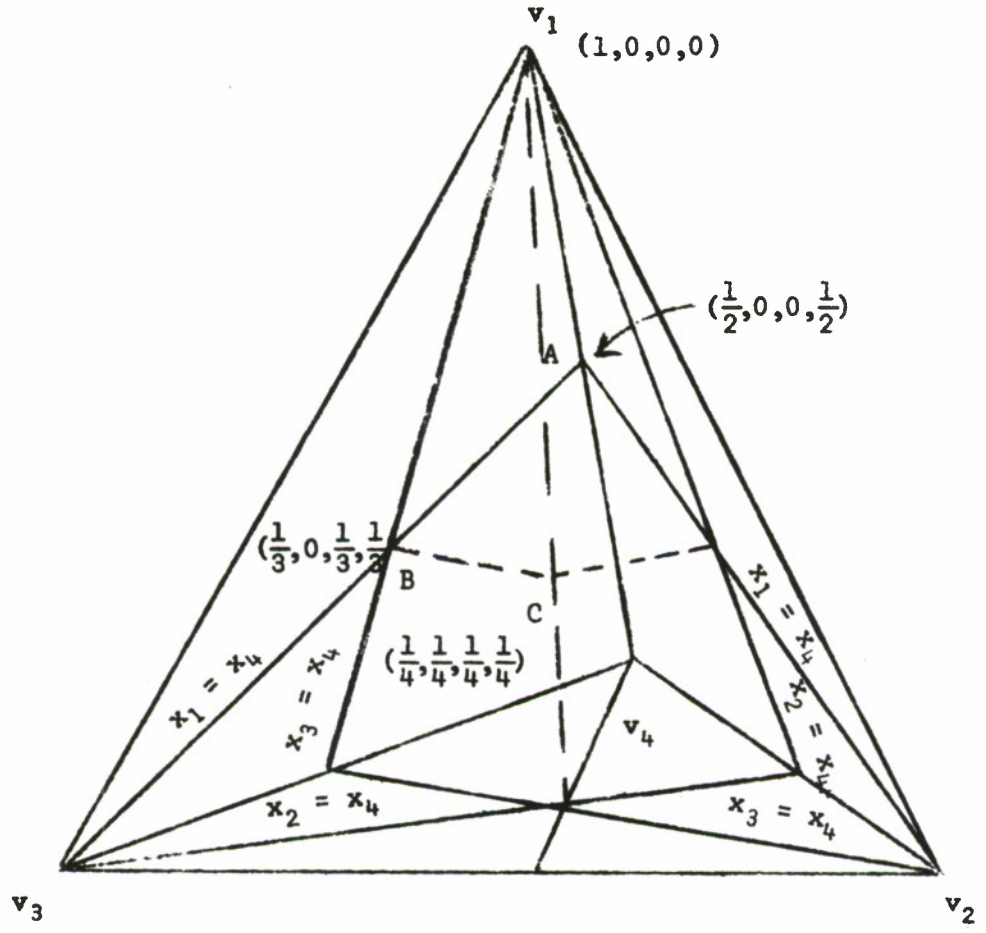


Figure 5. Four-dimensional Region Seen in Hyperplane of a Face, Showing Intersection of Four-dimensional Region with Face in Tetrahedron v_1 ABC

It is also the case that all the hyperplanes which intersect a given kth order face intersect in a point which is the center of the face. This can be seen by considering the particular kth order face determined by the vertices $(v_{i_1}, \ldots, v_{i_k})$. The hyperplanes which intersect this face are given by the equations

$$x_{i_1} = x_{i_2}, \quad x_{i_1} = x_{i_3}, \ \ldots, \ x_{i_1} = x_{i_k}$$

$$x_{i_2} = x_{i_3}, \ \ldots, \ x_{i_2} = x_{i_k}$$

$$\vdots$$

$$x_{i_{k-1}} = x_{i_k}$$

all of which are satisfied by the point

$$x_{i_1} = x_{i_2} = \cdots = x_{i_k} = \frac{1}{k}.$$

This point is the <u>center</u> of the kth order face.

We can now see that a region defined by this set of hyperplanes will be defined as an (n-1)-dimensional region determined by a series of vertex points which can be described as a sequence of midpoints of successive kth order faces, k going from 1 to n. This figure is a simplex in $n-1$ dimensions, as can be seen for the 4-dimensional case in Figure 5. The region is a tetrahedron, and the vertex points are v_1--a 1st order face, A--the midpoint of the 2nd order face (v_1, v_4), B--the midpoint of the 3rd order face (v_1, v_4, v_3), and C--the midpoint of the 4th order face.

With this in mind, we can count the number of regions determined in an n-dimensional octahedron by the set of hyperplanes $\{x_i = x_j\}$. If we examine the sequence of vertices defining the region in a combinatorial sense, we see that we can choose first any vertex v_i. Next, we must have a midpoint of an edge, with the constraint that the edge include the vertex v_i chosen previously. In the same way the kth vertex point

of the region is the midpoint of a kth order face which must contain the (k-1)st order face whose midpoint was chosen for the (k-1)st vertex point.

Denoting the n vertices of the region by $r_1, \ldots, r_n$, we see that we have n choices for r_1, $n - 1$ choices for r_2, ..., and only 1 choice for r_n. There are thus

$$n \cdot (n - 1) \cdot (n - 2) \cdot (n - 3) \cdots (1) = n!$$

regions.

The problem of partitioning the principal section into regions can be treated more generally by considering a set H of hyperplanes in n dimensions defined by the m equations

$$a_0 + a_1^1 x_1 + \cdots + a_n^1 x_n = 0$$

$$a_0 + a_1^m x_1 + \cdots + a_n^m x_n = 0$$

with the following restrictions

1. at least one $a_i^1 = 0$ $i = 1, \ldots, n$
2. the ordered set $(a_i^k$ $k = 1, 2, \ldots, n)$ is a permutation of the ordered set $(a_i^1$ $i = 1, 2, \ldots, n)$ for $k = 1, 2, \ldots, m$
3. m = the number of distinct permutations of $(a_i^1$ $i = 1, 2, \ldots, n)$
4. the line $x_1 = x_2 = \cdots = x_n$ is contained in every hyperplane.

In the hyperplane of the nth order face F H becomes a set H' of hyperplanes in $n - 1$ dimensions intersecting in the point

$$x_1 = x_2 = \cdots = x_n = \frac{1}{n},$$

the center C of the nth order face. H' defines a set of simplexes in F all of which have C as a vertex. By symmetry, all of these simplexes are congruent, and the (n-1)st order solid angle at C is equal for each of them. The sum of these angles is the complete (n-1)st order solid angle at C, so that the number P of regions is equal

to the complete (n-1)st order solid angle S_{n-1} at C divided by the (n-1)st order solid angle V_{n-1} determined at C by a single (n-1)-dimensional simplex formed from n of the hyperplanes in H'. In symbols,

$$p = \frac{S_{n-1}}{V_{n-1}} .$$

S_{n-1} is the surface content of the unit hypersphere in n - 1 dimensions,

$$S_{n-1} = \frac{2\pi^{\frac{n-1}{2}}}{\Gamma\left(\frac{n-1}{2}\right)}$$

$$= \frac{2\pi^k}{(k-1)!} \qquad \text{for } (n-1) \text{ even } (= 2k)$$

$$= \frac{2\pi^k k!\, 2^{2k}}{(2k)!} \qquad \text{for } (n-1) \text{ odd } (= 2k+1).$$

V_{n-1} may be determined by considering the simplex cut out in the hypersphere by the n (n-1)-dimensional hyperplanes forming the given (n-1)st order solid angle. This simplex may be considered as a simplex in an (n-2)-dimensional spherical space, and the problem is reduced to finding the volume of this simplex.

One way that this can be done is to transform the equations from the n-dimensional cartesian coordinate system $\{x_1,\ldots,x_n\}$ to an n-dimensional cylindrical system $\{z, r, \theta_1,\ldots,\theta_{n-2}\}$, with the z-axis on the line $x_1 = x_2 = \cdots = x_n$. The resulting expressions with $z = 1/\sqrt{n}$, $r = 1$ will give the intersection of the set of hyperplanes with the unit hypersphere circumscribing the nth order face. A simplex may be identified and its volume computed by techniques peculiar to the particular set $\{a_k^1\}$ chosen to define the set H of hyperplanes.

APPENDIX II

EXAMPLE OF USE OF G-SET ADAPTATION EQUATIONS

We will demonstrate the usefulness of the vector formulations of the g-set adaptation equations by solving the problem raised in Section IV concerning the determination of the configuration of the g-set when the output becomes zero during negative adaptation, using the adaptation equation developed in (5.26).

Specifically, we wish to find $\vec{g}(t)$ such that

$$\vec{x} \cdot \vec{g}(t) = 0 \tag{1}$$

From (5.26), we can expand $\vec{g}(t)$, so that (1) becomes

$$\vec{x} \cdot \vec{g}(0) + \frac{R(0)}{M_I \delta} \left(e^{F_I M_I \delta t} - 1\right) \vec{x} \cdot \vec{d} = 0 \tag{2}$$

But, from (4.21),

$$\vec{x} \cdot \vec{d} = \delta \tag{3}$$

and from (5.47), with $B = 0$,

$$\vec{x} \cdot \vec{g}(0) = \frac{R(0)}{M_I} \tag{4}$$

so that (2) becomes

$$\frac{R(0)}{M_I} e^{F_I M_I \delta t} = 0 \quad . \tag{5}$$

Since $F_I < 0$, this will occur when $t = \infty$, therefore

$$e^{F_I M_I \delta t} = 0 \quad . \tag{6}$$

Substituting (6) into the expression (5.26) for $\vec{g}(t)$, we have the solution:

$$\vec{g}(t) = \vec{g}(0) - \frac{R(0)}{M_I \delta} \vec{d} \quad . \tag{7}$$

In the numerical example from Table 4.2, we have

$$\vec{g}(0) = \begin{pmatrix} .5 \\ .5 \\ -.5 \\ -.5 \end{pmatrix} \qquad \vec{d} = \begin{pmatrix} -.1 \\ .1 \\ .2 \\ -.2 \end{pmatrix} \tag{8}$$

$$M_I = 1, \quad \delta = .1, \quad R(0) = .1,$$

so that

$$\frac{R(0)}{M_I \delta} = 1 \tag{9}$$

and

$$\vec{g}(t) = \begin{pmatrix} .6 \\ .4 \\ -.7 \\ -.3 \end{pmatrix} \tag{10}$$

This is the correct answer, since

$$\vec{x} \cdot \vec{g}(t) = .24 + .24 - .42 - .6 = 0 \tag{11}$$

APPENDIX III

SOLUTION TO THE SYSTEM OF DIFFERENTIAL EQUATIONS $\vec{y}' = \vec{u}\,\vec{v}^{t}\,\vec{y} + b\vec{u}$

The system of n simultaneous differential equations

$$\begin{aligned}
y_1' &= u_1v_1y_1 + u_1v_2y_2 + \cdots + u_1v_ny_n + bu_1 \\
y_2' &= u_2v_1y_1 + u_2v_2y_2 + \cdots + u_2v_ny_n + bu_2 \\
&\vdots \\
y_n' &= u_nv_1y_1 + u_nv_2y_2 + \cdots + u_nv_ny_n + b_1u_n
\end{aligned}$$

can be conveniently represented in the form

$$\vec{y}' = A\,\vec{y} + b\,\vec{u} \qquad (1)$$

where

$$y' = \begin{pmatrix} y_1' \\ y_2' \\ \cdot \\ \cdot \\ \cdot \\ y_n' \end{pmatrix} \qquad y = \begin{pmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_n \end{pmatrix}$$

and

$$A = \begin{pmatrix} u_1v_1 & u_1v_2 \cdots & u_1v_n \\ u_2v_1 & u_2v_2 \cdots & u_2v_n \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ u_nv_1 & u_nv_2 \cdots & u_nv_n \end{pmatrix} = \vec{u} \odot \vec{v}^{t}$$

$$\vec{u} = \begin{pmatrix} u_1 \\ u_2 \\ \cdot \\ \cdot \\ \cdot \\ u_n \end{pmatrix} \qquad \vec{v} = \begin{pmatrix} v_1 \\ v_2 \\ \cdot \\ \cdot \\ \cdot \\ v_n \end{pmatrix}$$

$\vec{v}^t$ is the transpose of $\vec{v}$, i.e.,

$$\vec{v}^t = (v_1, v_2 \ldots v_n)$$

and the operation $\odot$ is the matrix product. The symbol $\odot$ will be omitted whenever possible.

The solution of this system is given by

$$\vec{y} = \vec{y}_{\text{homogeneous}} + \vec{y}_{\text{particular}} \tag{2}$$

where $\vec{y}_{\text{homogeneous}}$ is the solution to the system

$$\vec{y}' = A\vec{y} \tag{3}$$

and $\vec{y}_{\text{particular}}$ is an additional particular solution to the non-homogeneous system.

For the homogeneous system, (3), we assume a solution of the form

$$\vec{y} = \vec{k}e^{\lambda x} \tag{4}$$

where k is a vector of constants. Substituting into (3), we obtain

$$\lambda\vec{k}e^{\lambda x} = A\vec{k}e^{\lambda x}$$

or

$$(A - \lambda I)\,\vec{k} = 0 \tag{5}$$

To obtain a solution other than $\vec{k} = 0$, the determinant

$$| A - \lambda I | = 0 \tag{6}$$

It should be noticed that (6) is the characteristic equation for A, having as solutions

$$\lambda = \lambda_1, \lambda = \lambda_2, \ldots, \lambda = \lambda_n$$

for $\lambda_1, \lambda_2, \ldots, \lambda_n$ the characteristic values of A. Associated with each λ_i is a vector of constants $\vec{k}_i$. Each $\vec{k}_i$ may be determined up to a single arbitrary constant c_i by substituting into (5) and solving the system

$$(A - \lambda_i I)\,\vec{k}_i = 0 \tag{7}$$

In order for (7) to hold, $\vec{k}_i$ must be a constant times the characteristic vector associated with λ_i. If $\vec{\alpha}_i$ is this characteristic vector,

$$\vec{k}_i = c_i \vec{\alpha}_i \tag{8}$$

For all λ_i distinct, (8) yields the complete solution for the system (3) by the superposition principle:

$$\vec{y} = c_1 \vec{\alpha}_1 e^{\lambda_1 x} + \cdots + c_n \vec{\alpha}_n e^{\lambda_n x}$$

If the λ_i are not distinct, but

$$\lambda_{i_1} = \lambda_{i_2} = \cdots = \lambda_{i_j} = \lambda_0,$$

then instead of (4) we must assume a solution of the form

$$\vec{y} = (\vec{k}_1 + \vec{k}_2 x + \cdots + \vec{k}_j x^{(j-1)})\, e^{\lambda_0 x} \tag{9}$$

In (3), the case at hand, we observe, following Wilf (1962), ch. 1, that $A = \vec{u} \odot \vec{v}^t$ is of rank 1, since for any vector $\vec{z}$

$$A\vec{z} = [\vec{u} \odot \vec{v}^t] \odot \vec{z} = \vec{u} \odot [\vec{v}^t \odot \vec{z}] = (\vec{v} \cdot \vec{z})\,\vec{u}, \tag{10}$$

$$\vec{v}^t \odot \vec{z} = v_1 z_1 + \cdots + v_n z_n = \vec{v} \cdot \vec{z} \tag{11}$$

($\vec{v} \cdot \vec{z}$ is the inner or dot product of the vectors $\vec{v}$ and $\vec{z}$.) (6), therefore, must reduce to

$$\lambda^{n-1}(\lambda - \lambda_0) = 0 \tag{12}$$

having as solutions

$$\lambda_1 = \lambda_0 , \lambda_2 = \lambda_3 = \cdots = \lambda_n = 0. \tag{13}$$

λ_0 may be evaluated by recalling that the trace of a matrix is equal to the sum of its characteristic values. (see Wilf (1962) In symbols:

$$tr(A) = u_1v_1 + u_2v_2 + \cdots + u_nv_n = \lambda_1 + \lambda_2 + \cdots + \lambda_n \tag{14}$$

Substituting (13) into (14) and noting (11), we have

$$tr(A) = \vec{u} \cdot \vec{v} = \lambda_0. \tag{15}$$

The characteristic vector $\vec{\alpha}_0$ associated with λ_0 may be obtained by observing that

$$A \odot \vec{u} = [\vec{u} \odot \vec{v}^t] \odot \vec{u} = (\vec{u} \cdot \vec{v})\vec{u} = \lambda_0\vec{u}$$

We have, therefore, that

$$\vec{\alpha}_0 = \vec{u}$$

and one solution to (3) is given by

$$\vec{y} = \vec{c}ue^{(\vec{u}\cdot\vec{v})x} = \vec{c}ue^{\lambda_0 x} \tag{16}$$

The repeated characteristic value 0 causes us to look for the remaining solution of the form (9) as

$$\vec{y} = (\vec{k}_1 + \vec{k}_2x + \vec{k}_3x^2 + \cdots + \vec{k}_{n-1}x^{(n-2)}) \tag{17}$$

The constants $\vec{k}_1 , \ldots , \vec{k}_{n-1}$ can be evaluated by substituting (17) into

$$\vec{y}' - A\vec{y} \equiv 0 \tag{18}$$

Differentiating (17), we obtain

$$\vec{y}' = \left[\vec{k}_2 + 2\vec{k}_3 x + \cdots + (n-2)\vec{k}_{n-1}x^{(n-3)}\right] \tag{19}$$

$$A\vec{y} = \vec{u} \cdot \vec{v}^t \cdot \vec{y} = \left[\vec{u} \quad (\vec{v}\cdot\vec{k}_1) + (\vec{v}\cdot\vec{k}_2)x + \cdots + (\vec{v}\cdot\vec{k}_{n-1})x^{(n-2)}\right] \tag{20}$$

Subtracting (19) from (20) and gathering terms,

$$A\vec{y} - \vec{y}' = \left[(\vec{v}\cdot\vec{k}_1)\vec{u} - \vec{k}_2\right] + \left[(\vec{v}\cdot\vec{k}_2)\vec{u} - 2\vec{k}_3\right]x \tag{21}$$

$$+ \cdots + \left[(\vec{v}\cdot\vec{k}_{n-2})\vec{u} - (n-2)\vec{k}_{n-1}\right]x^{n-3} + \left[(\vec{v}\cdot\vec{k}_{n-1})\vec{u}\right]x^{n-2} \equiv 0$$

Since the functions $1, x, x^2, \ldots, x^{n-1}$ are linearly independent (see [2]), we have the following n-2 equations by setting equal to zero each coefficient of t^j $j = 0, 1, \ldots, n-3$

$$\begin{aligned}
\vec{k}_2 &= (\vec{v}\cdot\vec{k}_1)\vec{u} \\
\vec{k}_3 &= \tfrac{1}{2}(\vec{v}\cdot\vec{k}_2)\vec{u} \\
&\cdot \qquad\qquad \cdot \\
&\cdot \qquad\qquad \cdot \\
&\cdot \qquad\qquad \cdot \\
\vec{k}_{n-1} &= \frac{1}{n-2}(\vec{v}\cdot\vec{k}_{n-2})\vec{u}
\end{aligned} \tag{22}$$

These equations are all dependent on $\vec{k}_1$, since they form a nested recursion relation. Solving them explicitly, they become:

$$\vec{k}_2 = (\vec{u} \cdot \vec{k}_1)\vec{u} \tag{23}$$

$$\vec{k}_3 = \frac{1}{2}[\vec{v} \cdot (\vec{v} \cdot \vec{k}_1)\vec{u}]\vec{u} = \frac{1}{2}(\vec{v} \cdot \vec{u})(\vec{v} \cdot \vec{k}_1)\vec{u} = \frac{1}{2}\lambda_0(\vec{v} \cdot \vec{k}_1)\vec{u}$$

.

.

.

$$k_{n-1} = \frac{1}{(n-2)\,!}\,\lambda_0^{n-3}\,(\vec{v} \cdot \vec{k}_1)\vec{u}$$

From the coefficient of $x^{(n-2)}$ in (21), we have the additional information that

$$(\vec{v} \cdot \vec{k}_{n-1})\vec{u} = 0 \tag{24}$$

$\vec{u} \neq o$ by hypothesis, so that we must have

$$\vec{v} \cdot \vec{k}_{n-1} = \vec{v} \cdot \left[\frac{1}{(n-2)!}\,\lambda_0^{n-3}\,(\vec{v} \cdot \vec{k}_1)\vec{u}\right] = \frac{\lambda_0^{n-2}}{(n-2)!}\,(\vec{v} \cdot \vec{k}_1) = 0 \tag{25}$$

By the same argument as in (24), we must have

$$\vec{v} \cdot \vec{k}_1 = 0 \tag{26}$$

Substituting (26) into (23), we obtain

$$\vec{k}_2 = \vec{k}_3 = \cdots = \vec{k}_{n-1} = 0$$

(17) therefore reduces to

$$\vec{y} = \vec{k}_1$$

and substituting into (18), we have

$$-A\vec{k}_1 = 0 \tag{27}$$

If we define $\vec{k}_1 = \vec{k}$, (27) may be solved

$$A\vec{k} = (\vec{u} \odot \vec{v}^t) \odot \vec{k} = (\vec{k} \cdot \vec{v})\vec{u} = 0$$

$$\vec{u} \neq 0 \quad , \text{ so that}$$

$$\vec{k} \cdot \vec{v} = 0$$

or
$$k_n = \frac{v_1 k_1 + \cdots + v_{n-1} k_{n-1}}{-v_n} \tag{28}$$

leaving n-1 arbitrary constants.

We now have the complete solution of (3) as

$$\vec{y} = c\vec{u}e^{\lambda x} + \vec{k}, \text{ for } \lambda = (\vec{u} \cdot \vec{v}) \tag{29}$$

The particular solution for the non-homogeneous system (1) is obtained by considering the system

$$\vec{y}' - A\vec{y} = b\vec{u} \tag{30}$$

taking

$$\vec{y} = Pb\vec{u}, \tag{31}$$

for P an $n \times n$ matrix, and substituting in (30), we have

$$-APb\vec{u} = b\vec{u}$$

$$-\vec{u} \odot \vec{v}^t \odot P \odot \vec{u} = \vec{u} \tag{32}$$

We observe that we can obtain a solution by taking

$$P = \frac{-(\vec{u} \odot \vec{v}^t)}{\lambda^2} \tag{33}$$

(32) becomes

$$-(\vec{u} \odot \vec{v}^t) \odot \frac{(-\vec{u} \odot \vec{v}^t)}{\lambda^2} \odot \vec{u} = \vec{u} \frac{(\vec{v} \cdot \vec{u})(\vec{v} \cdot \vec{u})}{\lambda^2} = \vec{u} \tag{34}$$

Therefore

$$\vec{y} = -\frac{b(\vec{u} \odot \vec{v}^t)}{\lambda^2} \vec{u} = -\frac{b\vec{u}}{\lambda} \tag{35}$$

is a particular solution of (1).

Substituting (29) and (35) into (2), we have the complete solution

$$\begin{aligned} \vec{y}_{\text{homogeneous}} &= \vec{c}\vec{u}e^{\lambda t} + \vec{k} \\ \vec{y}_{\text{particular}} &= \frac{-b}{\lambda} \vec{u} \end{aligned} \tag{36}$$

and

$$\vec{y} = \vec{c}\vec{u}e^{\lambda t} + \vec{k} - \left(\frac{b}{\lambda} \vec{u}\right) \tag{37}$$

APPENDIX IV

DERIVATION OF SURFACE ELEMENT FOR A HYPERSPHERE

The surface content of an n-dimensional hypersphere is a volume in n-1 dimensions. According to Somerville (1958), this volume is

$$S_n = \frac{2\pi^{\frac{n}{2}} r^{n-1}}{\Gamma\left(\frac{n}{2}\right)} \tag{1}$$

where r is the radius of the hypersphere and Γ is the gamma function, the generalized factorial. Note that

$$\Gamma(x) = (x - 1)\,\Gamma(x - 1)$$

and (2)

$$\Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}$$

To find the equation for a volume element, we must consider the geometry of the space. An (n-1)-dimensional hypersphere has the property that if a point be chosen on it as an origin, and a sequence of n-1 orthogonal great circles drawn, these circles will intersect again at a point diametrically opposite the origin, the line connecting the two points passing through the center of the hypersphere. These two points will be termed respectively, the North and South poles of the hypersphere.

We will use a spherical coordinate system for computing the volume, with n coordinates: a radius ρ and n-1 angles, $\phi_1,\ldots,\phi_{n-1}$. As in 3-dimensional spherical coordinates, the angles ϕ_i are of different kinds. In 3 dimensions, ϕ_1 is a conical angle from the North pole and ϕ_2 is a circular angle in a plane parallel to the equatorial plane, as illustrated in Figure 1. If ρ is fixed, x is constrained to lie on the sphere of radius ρ and with poles S and N. If ϕ_1 is then fixed, x is constrained to lie on the circle C, of radius $\rho \sin\phi_1$, the intersection of the sphere determined by ρ and the cone determined by ϕ_1. Fixing ϕ_2 then determines the exact location of the point on C. Angle ϕ_1 is measured from the line ON, and angle ϕ_2 is measured from the plane determined by the line ON and the great circle L_1 - more directly from the point on C given by the intersection of C and L_1. Note that ϕ_1, as a conical angle, ranges from 0 to π only, while ϕ_2, as a planar angle, ranges from 0 to 2π.

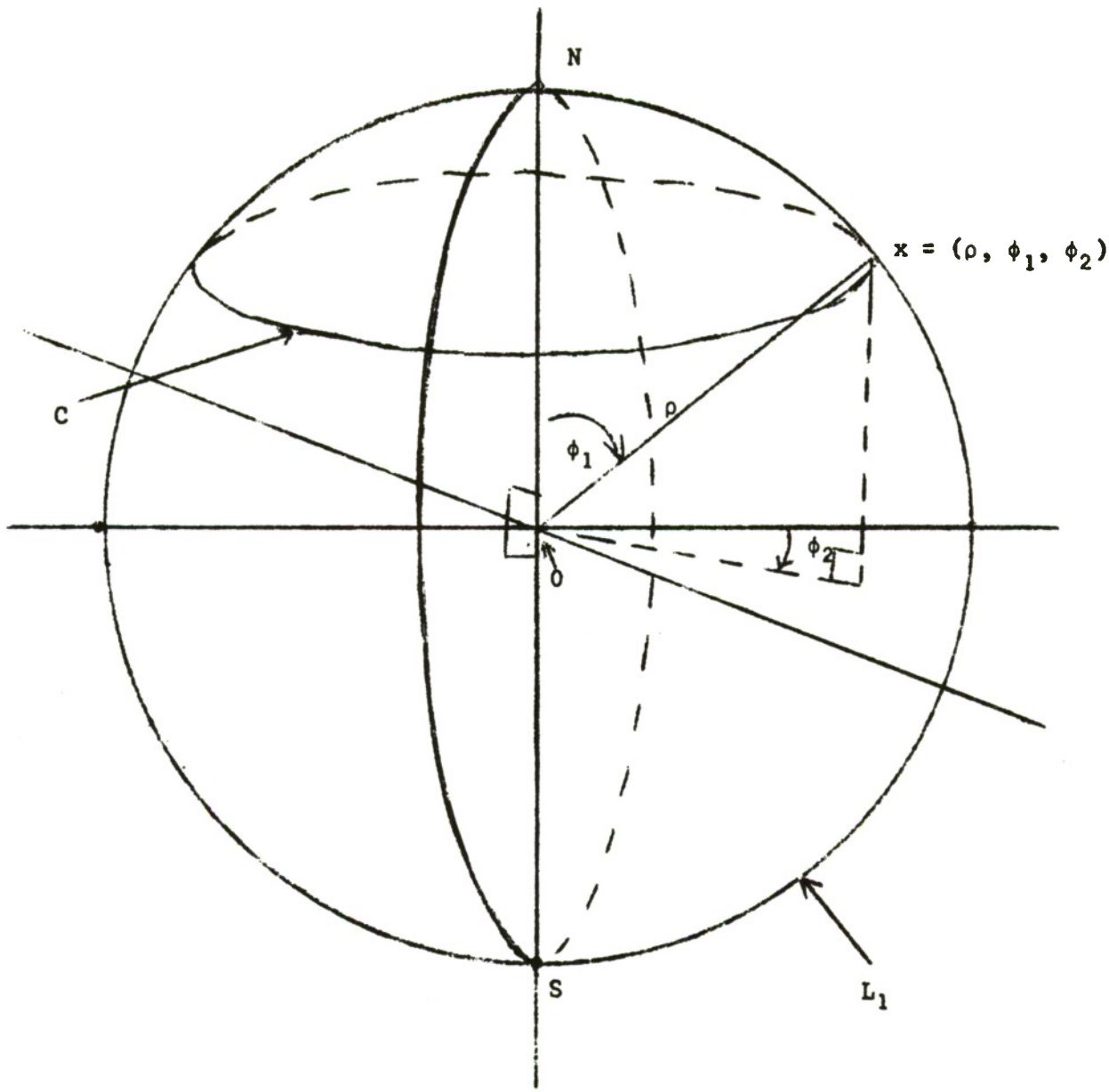


Figure 1. Three-dimensional Spherical Coordinate System

Figure 1. 3-Dimensional Spherical Coordinate System

This 3-dimensional system can be extended to n-dimensions in a straight forward manner. Fixing ρ determines a hypersphere. Fixing ϕ_1 determines an n-dimensional hypercone, the locus of all points x such that the line Ox is at an angle ϕ_1 with the line ON. The intersection of the two is clearly an n-1 dimensional hypersphere with radius $\rho\sin\theta_1$. If we select arbitrarily a great circle L_1 on the n-dimensional hypersphere, the intersection of L_1 with the (n-1)-dimensional hypersphere gives us a North pole for the latter hypersphere.

We can continue this process inductively, defining successive $(n - i + 1)$-dimensional hyperconical angles which restrict x to successive $(n - i)$-dimensional hyperspheres with radius $\rho\sin\phi_1 \sin\phi_2 \cdots \sin\phi_i$, until we come to ϕ_{n-2}, (which corresponds to the three-dimensional angle ϕ_1 of Figure 1). Angle ϕ_{n-1} is then a planar angle. We define the North pole of the $(n - i)$-dimensional hypersphere by its intersection with line (great circle) L_i of the $(n - i + 1)$-dimensional hypersphere. The successive surfaces L_1, (L_1, L_2), $(L_1, L_2, L_3),\ldots,(L_1,\ldots,L_{n-2})$ are coordinate surfaces imbedded in the original n-dimensional hypersphere.

A surface content element of the n-dimensional hypersphere is thus

$$dS_n = (\rho d\phi_1)(\rho\sin\phi_1 d\phi_2) \cdots (\rho\sin\phi_1 \cdots \sin\phi_{n-2} d\phi_{n-1})$$

and the allowed intervals are $0 < \phi_i < \pi$ for the hyperconical angles $\phi_1,\ldots, \phi_{n-2}$ and $0 < \phi_{n-1} < 2\pi$ for the planar angle.

APPENDIX V

CONTROL SYSTEM FORMULATION OF NEUROMIME INTERACTION

A. Introduction

Problems of interacting outputs in neuromime nets are of such a nature to lend themselves to analysis with control system techniques. This note gives a control system formulation of the S input computations of neuromimes. Figure 1 may be compared with Figure 2.5 of the text and the structures seen to be of the same kind.

B. Description of Model

Figure 1 is a block diagram of a two-element neuron net model. Standard control engineering notation is employed, wherein r's represent input time functions and c's represent output time responses. Each element has a forward gain of unity and an output which interacts with each other element by furnishing an additional input to each other element of a magnitude dictated by an "interaction coefficient" (K_{12} and K_{21} in Figure 1). This form of model can, of course, be generalized to an n-element net with n(n-1) interaction paths.

The responses (c's) of this model to a set of inputs (r's) is controlled by the values of the interaction coefficients. The analysis to follow will investigate this relationship between the neuron output and the degree and form of interaction associated with the other neuron outputs. The two-element model will be employed throughout since it is the least cumbersome to handle mathematically and produces conclusions which also hold for an n-element net.

C. Stability Criteria

The system equations for the two-element model of Figure 1 are

$$c_1 = r_1 + K_{12}c_2 \tag{1}$$

$$c_2 = r_2 + K_{21}c_1 \tag{2}$$

In matrix notation this equation set is expressed as

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 0 & K_{12} \\ K_{21} & 0 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \tag{3}$$

or

$$\begin{bmatrix} 1 & -K_{12} \\ -K_{21} & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \tag{4}$$

The output/input relationships obtained from simultaneous solution of Equations (1) and (2) are

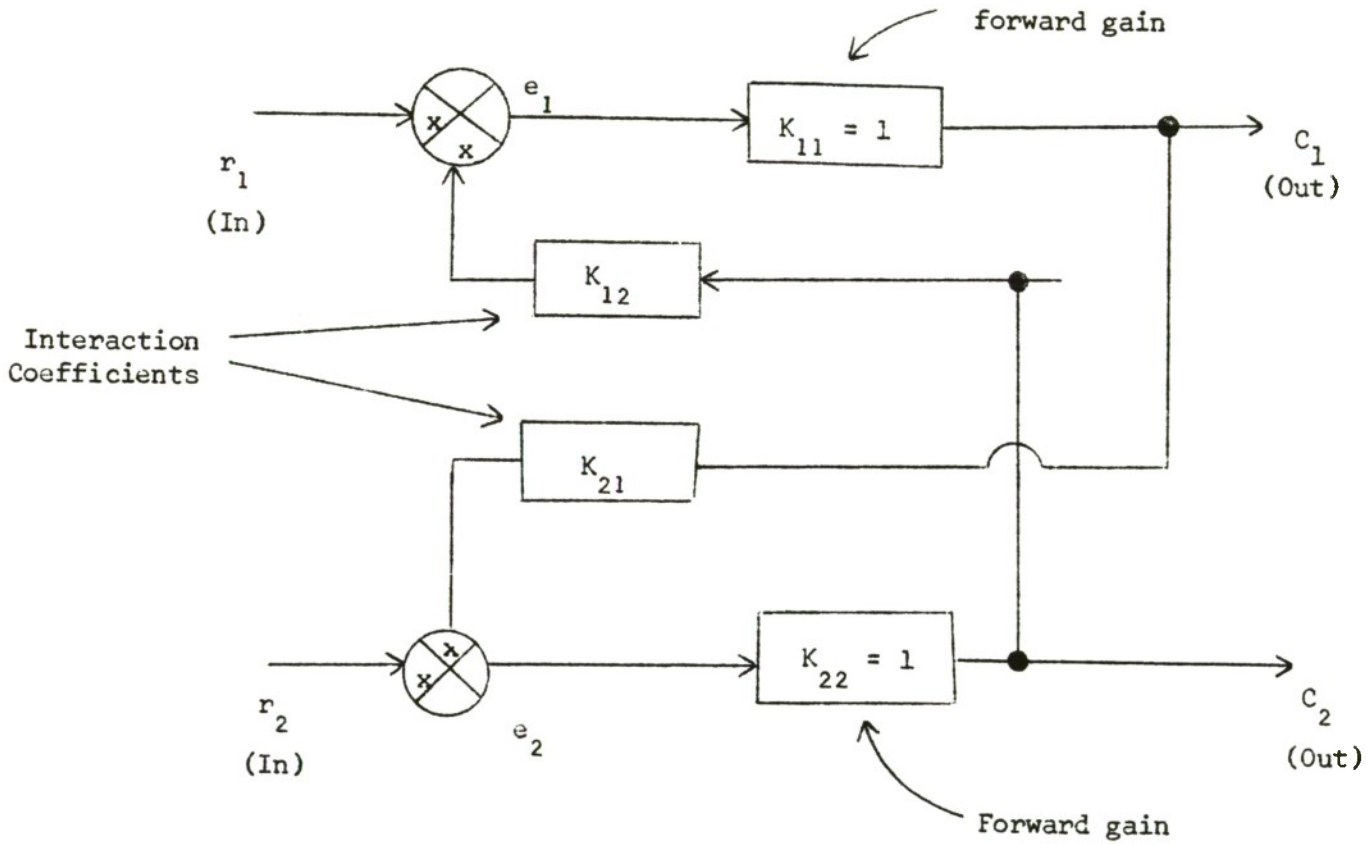


Figure 1. Block Diagram of a Two-element Neuron Net Model

$$\frac{c_1}{r_1} = \frac{1 + K_{12}(r_2/r_1)}{1 - K_{12}K_{21}} \tag{5}$$

and

$$\frac{c_2}{r_2} = \frac{1 + K_{21}(r_1/r_2)}{1 - K_{12}K_{21}} \tag{6}$$

According to Clark (1962), Equations (5) or (6) reveal that the outputs will increase without bound for any condition where the denominator, $1 - K_{12}K_{21}$, is equal to or less than zero. In terms of the interaction coefficients, the stability constraint is that the product of the interaction coefficients be less than +1.

This stability criteria can be more concisely expressed by referring to the matrix form of the system equations, Equation (4). The same stability constraint can be specified in terms of the determinant of the coefficient matrix, i.e.,

$$\begin{vmatrix} 1 & -K_{12} \\ -K_{21} & 1 \end{vmatrix} \geq 0 \ . \tag{7}$$

For an n-element neuron net model the matrix form of the system equations becomes

$$\begin{bmatrix} 1 & -K_{12} & -K_{13} & \cdots \\ -K_{21} & 1 & -K_{23} & \cdots \\ -K_{31} & -K_{32} & 1 & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \tag{8}$$

and hence the stability criteria is

$$\begin{bmatrix} 1 & -K_{12} & -K_{13} \cdots \\ -K_{21} & 1 & -K_{23} \cdots \\ -K_{31} & -K_{32} & 1 \cdots \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix} \geq 0 .$$

We emphasize that this stability criteria is in no way dependent on the inputs; only the signs and magnitudes of the interaction coefficients determine the stability of the output responses. Of course, for any stable set of interaction coefficients the output response will be directly affected by the inputs.

D. Performance of Refined Model

In real physiological systems transmission times are finite. To extend the utility of our basic neuron net model we are behooved to incorporate this physiological fact.

We have found the Laplace transform to be a very useful tool. (1) It enables us to obtain closed form expressions relating output to input for any type of feedback system. (2) It affords easy handling of systems which contain time delays.

Our initial efforts to incorporate these refinements of the neuron model have produced some interesting results. For instance, for a model containing neuron elements of +1 forward gain and positive interaction coefficients less than unity:

1. A set of step function inputs results in a set of outputs which are increasing, convergent "staircase" functions;

2. The final values of the staircase functions are determined by the magnitudes of the interaction coefficients (The final value may be ascertained from an equation of the form of (5) of Section C);

3. The time rates of convergence (time between steps of staircase) are dependent on the magnitudes of the transmission time delays associated with the forward paths of the neurons and the interaction paths between neurons.

A typical example illustrating these results for a neuron net of two elements is presented in Figure 2.

When values of +1 are assigned to all the interaction coefficients, the staircase response function increases without bound -- just as predicted by the stability criteria formulated in Section C.

E. Recommendations for Further Study and Refinement of Model

Our initial study of the performance characteristics of this refined neuron model indicate that it may be quite useful in aiding the understanding of phenomena associated with neural networks such as 1) Summation, 2) Variability and modification of response, and 3) the independence of the rhythms of stimulus and response.

Further study should involve:

1. Investigating model response for rectangular pulse inputs, impulse inputs, and input sets which contain both positive and negative inputs;

2. Further study of pertinent physiological phenomena to determine what additional refinements might be profitably incorporated.

Preconditions

Inputs: Positive unit step functions; forward gains: +1 (with 10 millisecond transmission time delays) interaction coefficients: $K_{12} = .5$, $K_{21} = .75$ (with 5 millisecond transmission time delays); total loop transmission time delay: 30 milliseconds

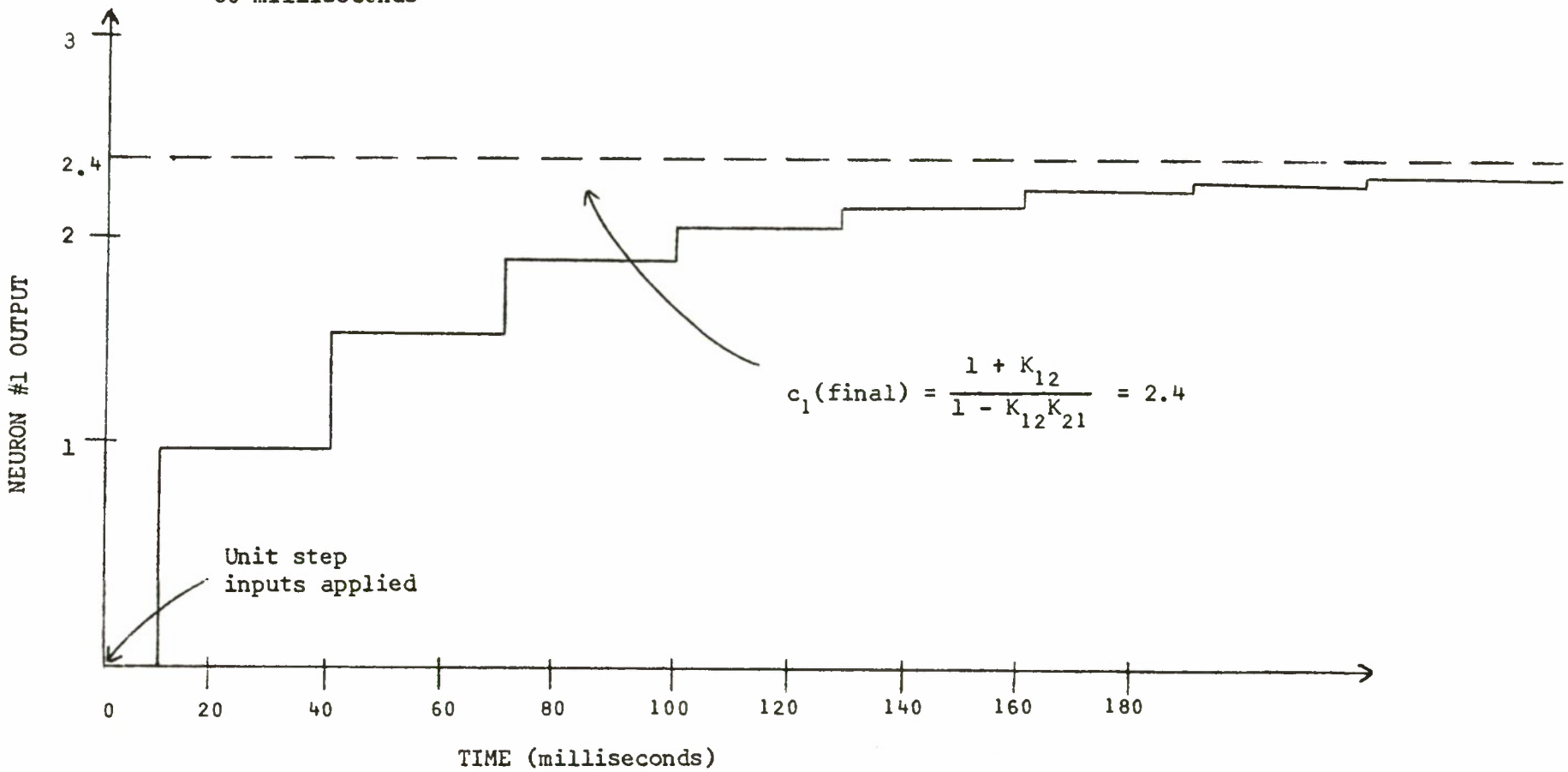


Figure 2. Typical Output Response for Unit Step Inputs in a Two-element Neuron Model with Transmission Time Delays and Positive Interaction Coefficients Less Than Unity

APPENDIX VI

COMPARISON OF ADAPTATION EQUATIONS FOR TWO NEUROMIME SIMULATORS

The adaptation equations of the two neuromime simulators are slightly different. The 7094 simulation program developed by Service Bureau Corporation[1] uses the equations discussed in the body of this report,

$$\frac{d}{dt}\vec{g}\,(t) = F \cdot R(t) \cdot (\vec{x} - \vec{\bar{x}}) \qquad (1)$$

$$= F \cdot R(t) \cdot \vec{d} \qquad \text{for } R(t) > 0$$

$$= 0 \qquad \text{for } R(t) \leq 0$$

The special purpose computer designed by Teledyne[2] uses the equations:

$$\frac{d}{dt}\vec{g}(t) = F \cdot [R] \cdot \vec{d} \qquad (2)$$

where

$$[R] = 1 \text{ if } R(t) > 0$$

$$[R] = 0 \text{ if } R(t) \leq 0$$

Note that both systems are zero when the output of the neuromimes is zero, which means that they will both adapt only to a positive simulus. The former, however, has a multiplicative factor R(t), while the latter has a factor of 1 when it is adapting. This means that system (2) will adapt at a constant rate independent of the output, while system (1) adapts at a rate which increases exponentially with time.

---

[1] Gracer, Franklin, and Orr, Kenneth, Neuromime Network Simulation, Final Report, Contract AF33(657)-11194, Service Bureau Corp., 14 August 1964.

[2] Neural Network Simulator, Final Engineering Report, Contract AF33(657)-8489, Teledyne Systems Co., January 1965.

A significant effect of this difference is that system (1) will have greater differential adaptation capability and less adaptation retention than system (2). Solving for $R(t)$ in (1) (See Section V of report), we have

$$R(t) = R(0)e^{FM_I\delta t} \tag{3}$$

where M_I is a constant and $\delta = |\vec{d}|^2$
and, solving (2), we have

$$R(t) = R(0) + F\delta t \tag{4}$$

From (3) and (4), one can see that if two neuromimes have inputs with the same δ, then the difference between their outputs will be constant in system (2), and will increase with system (1) if $F > 0$. If < 0, the difference will decrease in system (1) and again remain the same in system (2).

REFERENCES

Allais, D.C. (1964), The Selection of Measurements for Prediction, (AD 456-770) Stanford Electronics Laboratories, Stanford, California.

Ashby, W.R. (1948), "Design for a Brain," Electronic Engineering, December 1948.

Clark, R.N. (1962), Introduction to Automatic Control Systems, Wiley, New York, N.Y.

Hoffman, K., and Kunze, R. (1961), Linear Algebra, Prentice Hall, Engelwood Cliffs, New Jersey..

Inselberg, A., Von Foerster, H., et al. (1963), Computation of Invariants in Continuous and Discrete Manifolds, AMRL-TDR-63-40, University of Illinois, Urbana, Illinois.

Rosenblatt, F. (1962), Principles of Neurodynamics, Spartan, Washington, D.C.

Somerville, D.M.Y. (1958), An Introduction to the Geometry of N Dimensions, Dover, New York, N.Y.

Wiener, N. (1949), Interpolation, Extrapolation and Smoothing of Stationary Time Series, MIT Press, Cambridge, Massachusetts.

Wiener, N. (1958), Nonlinear Problems in Random Theory, MIT Press, Cambridge, Massachusetts.

Wilf, H.S. (1962), Mathematical Methods for the Physical Sciences, Wiley, New York, N.Y.

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)	2a. REPORT SECURITY CLASSIFICATION
Systems Research Laboratories, Inc. 500 Woods Drive Dayton, Ohio 45432	UNCLASSIFIED
	2b GROUP N/A

3. REPORT TITLE

INFORMATION HANDLING PROPERTIES OF NEUROMIME NETS

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Final report, 15 May 1964 - 31 May 1965

5. AUTHOR(S) (Last name, first name, initial)

Colomb, Robert M.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
September 1966	114	10

8a. CONTRACT OR GRANT NO. AF 33(615)-1825	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO. 7233	
c. Task No. 723302	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	AMRL-TR-66-128

10. AVAILABILITY/LIMITATION NOTICES

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Aerospace Medical Research Laboratories, Aerospace Medical Div., Air Force Systems Command, Wright-Patterson AFB, Ohio

13. ABSTRACT

This report is a study on some elementary information handling properties of neuromime nets, giving most emphasis to the functioning of a single neuromime component, and containing some discussion of the operation of simple nets. Single component computation is treated from the point of view of changes brought about in the internal structure by operations performed during data flow. A geometrical model is presented which illustrates the pattern measurement behavior of the component, and some of the simpler differential equations of adaptation are solved to provide some insight into the effect and interaction of the component control parameters. Simple net behavior is concerned mainly with feedback interaction among components, and gives some useful notation for describing net operation.

DD FORM 1 JAN 64 1473

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Neuromime nets Bionics Artificial intelligence Learning machines Mathematical						

INSTRUCTIONS

i. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security ciassification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulationa.

2*b*. GROUP: Automatic downgrading is specified in DoD Directive 5200.i0 and Armed Forces Industrial Manuai. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report tltie in all capital lettera. Tities in all casea ahould be unclaasified. If a meaningful title cannot be selected without ciassification, show titie ciassificat!on in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annuai, or final. Give the inciusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principai author is an absoiute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7*a*. TOTAL NUMBER OF PAGES: The total page count shouid follow normal pagination procedures, i.e., enter the number of pages containing information.

7*b*. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8*b*, 8*c*, & 8*d*. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9*b*. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator* or *by the sponsor)*, also enter this number(s).

i0. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security ciassification, using standard statementa auch as:

(1) "Quailfied requestera may obtain copies of thia report from DDC."

(2) "Forelgn announcement and dissemination of thia report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shali request through

_____."

(4) "U. S. miiitary agencies may obtain copies of this report directly from DDC. Other quaiified uaers shall request through

_____."

(5) "All distribution of this report ia controlled. Qualified DDC users shall request through

_____."

If the report has been furnished to the Office of Technicai Services, Department of Commerce, for aale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmentai project office or laboratory sponaoring *(paying for)* the research and development. Inciude address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet ahall be attached.

It is highly deslrable that the abstract of classified reporta be unclassified. Each paragraph of the abstract shall end with an indication of the miiltary aecurity classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

i4. KEY WORDS: Key words are technically meaningfui terms or short phrases that characterize a report and may be used aa index entries for cataloging the report. Key words must be selected so that no security classificatlon is required. Identifiers, such as equipment model designation, trade name, mlitary project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.